



Ministério da
**Ciência, Tecnologia
e Inovação**



sid.inpe.br/mtc-m19/2013/11.09.17.42-TDI

**FONTE DE LUZ ESPECTRALMENTE SINTONIZÁVEL
BASEADA EM LEDS PARA CALIBRAÇÃO DE SENSOR
DE ESTRELAS, COM APLICAÇÃO DO PARADIGMA
DE COMPUTAÇÃO EVOLUTIVA**

Marcos Eduardo Gomes Borges

Dissertação de Mestrado do Curso de Pós-Graduação em Computação Aplicada, orientada pelo Dr. Larmartine Nogueira Frutuoso Guimarães aprovada em 19 de dezembro de 2013.

URL do documento original:

<<http://urlib.net/8JMKD3MGP7W/3F7AKG2>>

INPE
São José dos Campos
2013

PUBLICADO POR:

Instituto Nacional de Pesquisas Espaciais - INPE

Gabinete do Diretor (GB)

Serviço de Informação e Documentação (SID)

Caixa Postal 515 - CEP 12.245-970

São José dos Campos - SP - Brasil

Tel.:(012) 3208-6923/6921

Fax: (012) 3208-6919

E-mail: pubtc@sid.inpe.br

CONSELHO DE EDITORAÇÃO E PRESERVAÇÃO DA PRODUÇÃO INTELLECTUAL DO INPE (RE/DIR-204):

Presidente:

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Membros:

Dr. Antonio Fernando Bertachini de Almeida Prado - Coordenação Engenharia e Tecnologia Espacial (ETE)

Dr^a Inez Staciarini Batista - Coordenação Ciências Espaciais e Atmosféricas (CEA)

Dr. Gerald Jean Francis Banon - Coordenação Observação da Terra (OBT)

Dr. Germano de Souza Kienbaum - Centro de Tecnologias Especiais (CTE)

Dr. Manoel Alonso Gan - Centro de Previsão de Tempo e Estudos Climáticos (CPT)

Dr^a Maria do Carmo de Andrade Nono - Conselho de Pós-Graduação

Dr. Plínio Carlos Alvalá - Centro de Ciência do Sistema Terrestre (CST)

BIBLIOTECA DIGITAL:

Dr. Gerald Jean Francis Banon - Coordenação de Observação da Terra (OBT)

REVISÃO E NORMALIZAÇÃO DOCUMENTÁRIA:

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Yolanda Ribeiro da Silva Souza - Serviço de Informação e Documentação (SID)

EDITORAÇÃO ELETRÔNICA:

Maria Tereza Smith de Brito - Serviço de Informação e Documentação (SID)

André Luis Dias Fernandes - Serviço de Informação e Documentação (SID)



Ministério da
**Ciência, Tecnologia
e Inovação**



sid.inpe.br/mtc-m19/2013/11.09.17.42-TDI

**FONTE DE LUZ ESPECTRALMENTE SINTONIZÁVEL
BASEADA EM LEDS PARA CALIBRAÇÃO DE SENSOR
DE ESTRELAS, COM APLICAÇÃO DO PARADIGMA
DE COMPUTAÇÃO EVOLUTIVA**

Marcos Eduardo Gomes Borges

Dissertação de Mestrado do Curso de Pós-Graduação em Computação Aplicada, orientada pelo Dr. Larmartine Nogueira Frutuoso Guimarães aprovada em 19 de dezembro de 2013.

URL do documento original:

<<http://urlib.net/8JMKD3MGP7W/3F7AKG2>>

INPE
São José dos Campos
2013

Dados Internacionais de Catalogação na Publicação (CIP)

Borges, Marcos Eduardo Gomes.

B644f Fonte de Luz espectralmente sintonizável baseada em LEDs para calibração de Sensor de Estrelas, com aplicação do Paradigma de Computação Evolutiva / Marcos Eduardo Gomes Borges. – São José dos Campos : INPE, 2013.

xxiv + 97 p. ; (sid.inpe.br/mtc-m19/2013/11.09.17.42-TDI)

Dissertação (Mestrado em Computação Aplicada) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2013.

Orientador : Dr. Lamartine Nogueira Frutuoso Guimarães.

1. sensor de estrelas. 2. simulador de estrelas. 3. fonte de luz multiespectral. 4. otimização de projeto multidisciplinar. 5 otimização extrema generalizado. I.Título.

CDU 681.5



Esta obra foi licenciada sob uma Licença [Creative Commons Atribuição-NãoComercial 3.0 Não Adaptada](https://creativecommons.org/licenses/by-nc/3.0/).

This work is licensed under a [Creative Commons Attribution-NonCommercial 3.0 Unported License](https://creativecommons.org/licenses/by-nc/3.0/).

Aprovado (a) pela Banca Examinadora
em cumprimento ao requisito exigido para
obtenção do Título de **Mestre** em
Computação Aplicada

Dr. Reinaldo Roberto Rosa



Presidente / INPE / SJC Campos - SP

Dr. Lamartine Nogueira Frutuoso
Guimarães



Orientador(a) / IEAv/CTA / SJC Campos - SP

Dr. Fabiano Luis de Sousa



Membro da Banca / INPE / SJC Campos - SP

Dr. Marco Antônio Pizarro



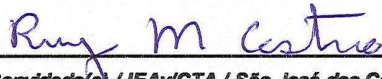
Membro da Banca / INPE / São José dos Campos - SP

Dr. Stephan Stephany



Membro da Banca / INPE / SJC Campos - SP

Dr. Ruy Morgado de Castro



Convidado(a) / IEAv/CTA / São José dos Campos - SP

Este trabalho foi aprovado por:

maioria simples

unanimidade



Aluno (a): **Marcos Eduardo Gomes Borges**

São José dos Campos, 19 de Dezembro de 2013

“Celui qui observe le vent ne sèmera point,
et celui qui regarde les nuages ne moissonnera point.”

“Quem observa o vento, nunca semeará,
e o que olha para as nuvens nunca segará.”

L'ECCLÉSIASTE, 11.04

A toda a minha família
e àqueles empenhados em melhorar o nosso país

AGRADECIMENTOS

Agradeço primeiramente a Deus pela oportunidade de vida, pela família que Ele me concedeu, e por me abençoar e me guiar em todas as escolhas difíceis da minha vida.

À minha eterna companheira e amiga Jaciara de Queiroz Borges, pelo seu apoio, compreensão nos momentos difíceis, paciência, amor e ajuda na revisão do texto.

À minha mãe Marcia Regina Gomes e a minha vovó Maria Sebastiana de Almeida por toda a educação, apoio e motivação.

Ao amigo Márcio Afonso Arimura Fialho e ao Dr. Mário Luiz Selingardi, pela liberação parcial das minhas atividades na Divisão de Eletrônica Aeroespacial do INPE, e por todo apoio recebido durante o desenvolvimento deste trabalho. Também agradeço a todos os colegas que direta ou indiretamente contribuíram na execução deste trabalho.

E por último, não menos importante, este trabalho não teria sido possível sem a valiosa orientação do Dr. Lamartine Nogueira Frutuoso Guimarães (INPE/IEAv) e das importantes contribuições do Dr. Bráulio Fonseca Carneiro de Albuquerque (INPE). A eles, meus sinceros agradecimentos.

RESUMO

Um Sensor de Estrelas Autônomo (SEA) foi projetado e construído no INPE. Para testar, caracterizar e calibrar o SEA é necessário uma infraestrutura de testes sofisticada, que inclui dentre diversos instrumentos, um Simulador de Estrelas (SE). O SE é formado por uma fonte de luz espectralmente sintonizável baseada em LEDs (do inglês *Light Emitting Diode*), alvo (*pinhole*) e um colimador parabólico fora de eixo. A fonte de luz é responsável em gerar espectros contínuos semelhantes aos emitidos por estrelas. O pinhole é o orifício por onde os feixes de luz gerados incidem e são projetados no colimador. O colimador tem a função de tornar paralelo o feixe luminoso divergente recebido através do pinhole simulando uma fonte de luz no infinito, ou seja, uma fonte de luz pontual semelhante a uma estrela. Durante o projeto e o desenvolvimento do SE foi necessário definir a quantidade e tipos de LEDs que seriam utilizados para conseguir simular estrelas necessárias para a calibração e caracterização do SEA. Um software foi desenvolvido para simular o comportamento da fonte de luz baseada em LEDs. A escolha ótima dos LEDs foi realizada utilizando o simulador da fonte de luz e um algoritmo de otimização multiobjetivo, o M-GEO. Após essa etapa os LEDs foram comprados e montados na fonte de luz; diversos estudos e simulações foram realizadas e um software em C++ foi desenvolvido para controlar a fonte de luz com a finalidade de simular estrelas. Sendo assim, este trabalho tem como principal objetivo apresentar a metodologia aplicada na definição de parâmetros de projeto do SE, no desenvolvimento de algoritmos para simular a fonte de luz e no desenvolvimento de software de controle do SE. A metodologia aplicada utiliza como ferramentas, paradigmas de Inteligência Artificial (IA) e algoritmos clássicos de otimização. Para a conclusão do projeto e da construção do SE, foi desenvolvido um sistema de controle em malha fechada utilizando dois métodos de otimização, o algoritmo de Levenberg-Marquardt (LM) e o Gradiente Descendente (GD), que se mostraram eficientes para, através da composição espectral dos diferentes LEDs em diferentes potências, gerar espectros similares aos de estrelas na faixa do visível e infravermelho próximo.

A LED-BASED SPECTRALLY TUNABLE LIGHT SOURCE FOR CALIBRATION OF STAR TRACKER, WITH THE APPLICATION OF EVOLUTIONARY COMPUTING PARADIGM

ABSTRACT

An Autonomous Star Tracker (AST) has been designed and constructed at the National Institute for Space Research (INPE). In order to test, characterize and calibrate the AST, a sophisticated testing infrastructure is needed. This setup is composed of several instruments including a Star Continuum Spectrum Simulator (SCSS), which was also developed at INPE. The SCSS is basically composed of a LED-based spectrally tunable light source, a pinhole and a collimator. The SCSS is designed to have the capability of producing different electromagnetic continuum spectral distributions, as those of the stars, for the AST optical calibration. The light source is responsible for generating continuum spectra similar to those emitted by stars. The pinhole is the orifice through which the generated light beams fall and are projected onto the collimator. The collimator then arranges in parallel the diverging light beams received through the pinhole, simulating a light source at infinity, like those emitted by stars. During the design and development of the SCSS it was necessary to define the amount and types of LEDs. One software was developed to simulate the behavior of the LED-based light source. The optimal choice of the LEDs was performed using the simulator of the light source and a multi-objective optimization algorithm, the M-GEO. The main goal of this work is to present the methodology used in the definition of design parameters of the SCSS. It Develops algorithms to simulate the light source and a software to control the SCSS. The methodology uses as tools, paradigms of Artificial Intelligence (AI) and classical optimization algorithms. Finally, to complete the design and construction of the SCSS, a feedback control system using two optimization methods, the Levenberg-Marquardt (LM) and Gradient Descent (GD) algorithms, were developed. These methods have proven efficient for simulating star necessary for the calibration of AST.

LISTA DE FIGURAS

	<u>Pág.</u>
2.1 Sensor de Estrelas Autônomo em desenvolvimento no INPE.	7
2.2 Configuração do Simulador de Estrelas construído no INPE.	7
2.3 Principais itens que compõem o Simulador de Estrelas.	8
2.4 Irradiância espectral de uma estrela de magnitude zero a 7500K.	11
2.5 Histograma de temperatura das estrelas de magnitudes zero a cinco, se- lecionadas do Catálogo de Estrelas Hipparcos.	12
2.6 Simulação da irradiância espectral de uma estrela no Simulador de Estrelas.	13
2.7 Energia emitida por unidade de área e ângulo sólido em uma dada direção.	13
2.8 DPE de dois LEDs modelados comparados com LEDs reais.	16
2.9 FWHM da distribuição de radiação com perfil gaussiano.	17
2.10 Ângulo de meia potência (θ_{fwhm} ou $2\theta_{\frac{1}{2}}$).	17
2.11 Soluções para um problema multiobjetivo de minimização, plotado no espaço de funções objetivo F1 e F2.	21
2.12 Variáveis de projeto codificadas em uma string binária; neste exemplo cada variável é representada por seis bits.	22
2.13 Fluxograma do algoritmo GEO.	25
2.14 Fluxograma do algoritmo M-GEO.	28
3.1 Fluxograma para modelagem dos LEDs.	30
3.2 Interface gráfica do Software em MATLAB utilizado para modelar os LEDs.	31
3.3 DPE de 160 LEDs candidatos.	31
3.4 Irradiância espectral de todas as estrelas de magnitude zero, de tempe- ratura de 3000K a 20000K selecionadas do catálogo de estrelas Hipparcos.	32
3.5 Irradiância espectral cobrindo todas as estrelas de magnitude zero, de temperatura de 3000K a 20000K selecionadas do catálogo de estrelas Hipparcos.	32
3.6 Ajuste de curva para a solução com 1202 LEDs conseguida com LM. . .	35
3.7 Ajuste de curva para a solução com 200 LEDs conseguida com LM. . . .	36
3.8 Erro de ajuste de curva médio, <i>fitness</i> , em 50 execuções independentes do algoritmo GEO, para identificar o mínimo global em função do τ . Critérios de parada: $ F(X)_{melhor} - F(X) \leq 1 \times 10^{-15}$ e $NAF > 1\,440\,000$.	39

3.9	Número médio de avaliações da função objetivo, NAF , em 50 execuções independentes do algoritmo GEO, para identificar o mínimo global em função do τ . Critérios de parada: $ F(X)_{melhor} - F(X) \leq 6 \times 10^{-15}$ e $NAF > 1\,440\,000$	40
3.10	Ajuste de curva para a solução com 500 LEDs conseguida com GEO.	41
3.11	Fronteira de Pareto gerada pelo M-GEO.	43
3.12	Ajuste de curva para a solução com 253 LEDs conseguida com M-GEO.	44
3.13	Ajuste de curva para a solução com 229 LEDs conseguida com M-GEO.	44
3.14	Ajuste de curva para a solução com 198 LEDs conseguida com M-GEO.	45
3.15	Ajuste de curva para a solução com 174 LEDs conseguida com M-GEO.	45
3.16	Simulação da FLS para uma estrela de magnitude zero a 7 500K.	46
3.17	Simulação da FLS para uma estrela de magnitude zero a 20 000K.	46
3.18	Simulação da FLS para uma estrela de magnitude zero a 5 600K.	47
4.1	Interface gráfica do Software da FLS, implementado para simular estrelas.	50
4.2	Simulação de estrela de magnitude zero a 3 500K conseguido com LM.	52
4.3	Simulação de estrela de magnitude zero a 5 800K conseguido com LM.	53
4.4	Simulação de estrela de magnitude zero a 12 130K conseguido com LM.	53
4.5	Simulação de estrela de magnitude zero a 3 500K conseguido com GD.	55
4.6	Simulação de estrela de magnitude zero a 5 800K conseguido com GD.	56
4.7	Simulação de estrela de magnitude zero a 12 130K conseguido com GD.	56
4.8	Comparação da convergência dos algoritmos LM e GD, em 50 execuções independentes para simular uma estrela de 3 500K.	57
4.9	Comparação da convergência dos algoritmos LM e GD, em 50 execuções independentes para simular uma estrela de 5 800K.	58
4.10	Comparação da convergência dos algoritmos LM e GD, em 50 execuções independentes para simular uma estrela de 7 300K.	58
4.11	Comparação da convergência dos algoritmos LM e GD, em 50 execuções independentes para simular uma estrela de 20 000K.	59
4.12	Variação da irradiância espectral da FLS para uma estrela de magnitude zero a 7 500K, com dados coletados durante sete dias.	60
4.13	Variação da irradiância da FLS para uma estrela de magnitude zero a 7 500K, com dados coletados durante sete dias.	60
C.1	Configuração inicial do SMS-500.	83
C.2	Software do Simulador de Estrelas: Módulo SMS500.	84
C.3	Caixa de diálogo para calibração do SMS-500.	86
C.4	Caixa de diálogo para seleção do arquivo de calibração do SMS-500.	86
C.5	Caixa de diálogo com procedimento inicial para calibração do SMS-500.	87
C.6	Caixa de diálogo com procedimento final para calibração do SMS-500.	87

C.7	Caixa de diálogo informando a finalização da calibração do SMS-500.	87
C.8	Software do Simulador de Estrelas: Módulo LED Driver.	89
C.9	Software do Simulador de Estrelas: Módulo Long Term Stability.	90
C.10	Software do Simulador de Estrelas: Módulo Star Simulator.	91
D.1	Configuração inicial do SMS-500.	94
D.2	Configuração das variáveis de usuário do Windows.	95
D.3	Configuração das variáveis de sistema do Windows.	95

LISTA DE ABREVIATURAS E SIGLAS

AG	–	Algoritmo Genético
AGS	–	Algoritmo Genético Simples
APS	–	Active Pixel Sensor
CCD	–	Charge Coupled Devices
CDS	–	Centre de Donnés Astronomiques de Strasbourg
CIE	–	Commission Internationale de l’Eclairage
DEA	–	Divisão de Eletrônica Aeroespacial do INPE
DPE	–	Distribuição de Potência Espectral
ESA	–	European Space Agency
FPGA	–	Field Programmable Gate Arrays
FLS	–	Fonte de Luz Espectralmente Sintonizável
FWHM	–	Full Width at Half Maximum
GEO	–	Generalized Extremal Optimization
GD	–	Gradiente Descendente
GSE	–	Ground Support Equipment
IA	–	Inteligência Artificial
IEAv	–	Instituto de Estudos Avançados
INPE	–	Instituto Nacional de Pesquisas Espaciais
LED	–	Light Emitting Diode
LM	–	Levenberg-Marquardt
M-GEO	–	Multiobjective GEO
NIST	–	National Institute of Standards and Technology
RS	–	Recozimento Simulado
SCD1	–	Satélite de Coleta de Dados 1
SE	–	Simulador de Estrelas
SEA	–	Sensor de Estrelas Autônomo
DLL	–	Dynamic-Link Library (Biblioteca de Vínculo Dinâmico)

LISTA DE SÍMBOLOS

A_c	– Área do feixe do colimador (porta de entrada da esfera integradora)
A_e	– Área da porta de saída da esfera integradora
A_i	– Área da porta de entrada da esfera integradora
A_p	– Área do <i>pinhole</i>
A_s	– Área da casca da esfera integradora
c	– Velocidade da luz (m/s)
C	– Cadeia binária usada para codificar X
$\Delta\lambda_{0,5}$	– Comprimento da metade da potência espectral (nm)
$\Delta F(X_i)$	– Diferença entre $F(X_i)$ e o valor de referência $F(X_{melhor})$
e_i	– Precisão desejada para X_i
E	– Irradiância (W m^{-2})
<i>fitness</i>	– Erro de ajuste de curva entre o espectro desejado e o obtido
$f/\#$	– Razão entre a distância focal e o diâmetro de entrada do colimador
$F(X)$	– Valor da função objetivo
$F(X_{melhor})$	– Valor da função objetivo para X_{melhor}
F_{transf}	– Função de transferência do sistema esfera integradora, <i>pinhole</i> e colimador
h	– Constante de Planck (J·s)
$I(\lambda)$	– Irradiância espectral da estrela ($\text{W m}^{-2} \text{nm}^{-1}$)
I_{550}	– Irradiância da estrela no comprimento de onda de 550 nm
$I(\theta_x, \theta_y)$	– Intensidade de radiação por ângulo de um LED
I_v	– Intensidade luminosa (cd)
k	– Constante de Boltzmann (J/K)
k	– Índice de adaptabilidade
K_m	– Máxima eficácia espectral luminosa para a visão (683 lm/W em 555 nm)
l_i	– Número de bits que codifica X_i
λ	– Comprimento de onda (nm)
λ_0	– Comprimento de onda de pico (nm)
L	– Radiância ($\text{W cm}^{-2} \text{sr}^{-1}$)
L	– Número total de bits usados na codificação das variáveis de projeto
m_v	– Magnitude visual
$M(\lambda)$	– Exitância espectral do corpo negro ($\text{W m}^{-2} \text{nm}^{-1}$)
NAF	– Número de avaliações de $F(X)$
$NFOBJ$	– Número de funções objetivo
ρ	– Fator de reflectância interna da esfera
Φ	– Fluxo
Φ_v	– Fluxo luminoso (lm)
Φ_e	– Fluxo radiante (W)
$P(k)$	– Probabilidade do bit de índice de adaptação k sofrer mutação
R_{550}	– Razão entre I_{550} e $M(550)$

RAN	–	Número aleatório
$S_{LED}(\lambda)$	–	Distribuição de potência espectral de um LED
T	–	Temperatura do corpo negro (K)
$Total_{LEDs}$	–	Número total de LEDs
τ	–	Parâmetro real não negativo
θ_{fwhm} ou $2\theta_{\frac{1}{2}}$	–	Ângulo de meia potência
$V(\lambda)$	–	Resposta espectral do olho humano
X	–	Vetor das variáveis de projeto
X_i	–	i -ésimo elemento de X
X_{melhor}	–	Melhor X encontrado durante a busca
X_{max}^i	–	Valor máximo de X_i
X_{min}^i	–	Valor mínimo de X_i

SUMÁRIO

	<u>Pág.</u>
1 INTRODUÇÃO	1
1.1 Objetivos e Metodologia	2
1.2 Motivação	3
1.3 Organização do Trabalho	4
2 FUNDAMENTOS TEÓRICOS	5
2.1 Sensor de Estrelas Autônomo (SEA)	5
2.2 Simulador de Estrelas (SE)	7
2.3 Modelagem da irradiação espectral das estrelas	9
2.3.1 Magnitude das estrelas	9
2.3.2 Irradiância espectral das estrelas	10
2.3.3 Irradiância espectral desejada na saída do colimador	12
2.4 Teoria e equações para modelagem dos LEDs	15
2.4.1 Distribuição de potência espectral de um LED	15
2.4.2 Distribuição de radiação de um LED por ângulo	16
2.4.3 Cálculo do fluxo luminoso de um LED	17
2.5 O Problema de Otimização	20
2.5.1 Otimização Extrema Generalizada (GEO)	22
2.5.2 Otimização Extrema Generalizada Multiobjetivo (M-GEO)	25
3 O PROBLEMA DA ESCOLHA ÓTIMA DOS LEDS	29
3.1 Processo de modelagem dos LEDs	29
3.2 Escolha dos LEDs com Levenberg-Marquardt	32
3.3 Escolha dos LEDs usando paradigmas de IA	37
3.3.1 Escolha dos LEDs com GEO	41
3.3.2 Escolha dos LEDs com M-GEO	42
4 O SOFTWARE DE CONTROLE DA FLS	49
4.1 O algoritmo de Levenberg-Marquardt (LM)	50
4.2 O algoritmo do Gradiente Descendente (GD)	54
4.3 Levenberg-Marquardt vs Gradiente Descendente	57
4.4 Estabilidade da simulação das estrelas	59

5 CONCLUSÃO	63
REFERÊNCIAS BIBLIOGRÁFICAS	67
APÊNDICE A - LEDs candidatos a serem utilizados na FLS.	73
APÊNDICE B - Rotinas em MATLAB.	79
APÊNDICE C - Manual do Software do Simulador de Estrelas	83
C.1 Instalação e configuração do software SphereOptics SMS-500	84
C.2 Módulo SMS500	84
C.2.1 Procedimento de calibração do SMS-500	85
C.2.2 Opções de configuração do módulo SMS500	88
C.3 Módulo LED Driver	89
C.4 Módulo Long Term Stability	90
C.5 Módulo Star Simulator	91
APÊNDICE D - Configuração do ambiente de desenvolvimento	93
D.1 Instalação e configuração do software SphereOptics SMS-500	93
D.1.1 Instalação e configuração do Qt 5.0 e Qwt 6.1	94

1 INTRODUÇÃO

O INPE tem se posicionado para ser o principal centro de pesquisa e desenvolvimento nacional nas áreas de espaço e observação da terra desde o início da Missão Espacial Completa Brasileira nos anos 80. O primeiro satélite desenvolvido, o SCD1, contava com sensores solares totalmente desenvolvidos no INPE, com a participação das universidades brasileiras. O desenvolvimento dos sensores terrestres e de estrelas foram iniciados mais tarde, com o objetivo de construir protótipos que comprovassem o domínio da tecnologia e que esta pudesse ser repassada para a indústria nacional (BERTOLINO, 2004).

Um Sensor de Estrelas é um equipamento eletro-óptico utilizado para determinar a atitude de veículos espaciais ou suborbitais utilizando como fonte de referência as estrelas. A determinação de atitude corresponde a identificar a região do espaço para a qual o sensor está apontando, e para isso, o sensor adquire e processa imagens com o intuito de determinar as estrelas presentes no campo de visada e assim calcular a atitude (WERTZ, 1978; WERTZ; LARSON, 1999).

O desenvolvimento do sensor de estrelas iniciou-se no INPE em 1997. O objetivo inicial foi construir um protótipo de laboratório sem as preocupações de redução de massa, volume, consumo de energia e qualificação espacial. O primeiro protótipo do sensor foi concluído em 2001 e testado no Balão MASCO em 2004 como equipamento auxiliar na determinação de atitude. Esse protótipo foi construído utilizando detectores do tipo CCD (do inglês *Charge Coupled Devices*), era do tipo não autônomo e informava apenas a posição angular e a magnitude das estrelas presentes no seu campo de visada.

A partir de 2001, uma nova fase de desenvolvimento foi iniciada, com o objetivo de construir um Sensor de Estrelas Autônomo (SEA) (BERTOLINO, 2004; ALBUQUERQUE, 2005; FIALHO, 2007). Para testar, caracterizar e calibrar o sensor de estrelas é necessário uma infraestrutura de testes sofisticada chamada GSE (do inglês *Ground Support Equipment*). O GSE é composto por diversos equipamentos eletrônicos, ópticos, eletro-ópticos, mecânicos e eletromecânicos. Dentre os diversos itens que compõem o GSE, destaca-se o Simulador de Estrelas (SE)¹, que é o equipamento responsável por gerar espectros contínuos semelhantes aos emitidos por estrelas.

Para que o SE consiga simular estrelas, ele utiliza uma Fonte de Luz Espectralmente

¹O termo **Simulador de Estrelas** é utilizado neste trabalho para referenciar um equipamento composto por hardware e software, capaz de simular a magnitude e o espectro contínuo de estrelas.

Sintonizável baseada em LEDs (FLS), um colimador parabólico fora de eixo e um *pinhole* em seu plano focal. A FLS é o equipamento principal do SE, e responsável por gerar os espectros contínuos semelhantes aos de estrelas. O *pinhole* juntamente com o colimador têm a função de projetar no infinito o espectro gerado pela FLS, dando a impressão visual e também radiométrica de uma fonte de luz pontual muito semelhante a uma estrela.

O SE é de extrema importância para a realização de testes e calibração do SEA. Atualmente duas versões de engenharia do SEA estão sendo construídas, e farão parte da malha de controle e atitude do experimento protoMIRAX, que visa testar subsistemas que podem ser reutilizados em missões de satélites do INPE. Além do objetivo tecnológico, o protoMIRAX também tem como objetivo coletar dados científicos, tais como imagens, espectros e curvas de variabilidade de fontes importantes de raios-x cósmicos.

1.1 Objetivos e Metodologia

A presente dissertação de mestrado tem como principal objetivo apresentar a metodologia desenvolvida utilizando paradigma de Inteligência Artificial (IA) aplicada no projeto e construção da FLS, bem como o desenvolvimento de um sistema de controle em malha fechada para que a fonte de luz gere espectros sintéticos de maneira estável.

O processo todo envolveu estudos na área de radiometria e fotometria, engenharia espacial e, em sua maior parte computacional. A FLS foi totalmente projetada e construída no INPE. Durante o seu projeto foi preciso definir de forma ótima a quantidade e tipos de LEDs necessários para gerar espectros de estrelas exigidos na calibração e testes do SEA. Nessa fase, não havia LEDs em quantidades e tipos suficientes para a realização da escolha dos que comporiam a FLS, ou seja, havia a necessidade de comprar uma grande quantidade de LEDs para a realização de medições e análises radiométricas para orientar a escolha adequada. Outra opção seria modelar as características espectrais dos LEDs e viabilizar a escolha dos mesmos via simulações em computador.

Devido as dificuldades e a morosidade no processo de compra, optou-se pelo desenvolvimento de um software em computador para simular o comportamento da FLS. Utilizando o simulador da FLS e modelando as características espectrais dos LEDs, e utilizando paradigma de IA foi possível realizar a escolha ótima dos LEDs por meio de simulações em computador.

Após definir a quantidade e tipos de LEDs, os mesmos foram comprados e montados na FLS. Nessa etapa, foi realizado o levantamento das características espectrais dos LEDs utilizando um espectroradiômetro para coleta das informações. Com os dados espectrais dos LEDs coletados, foi implementado um sistema de controle com retroalimentação capaz de controlar a FLS e assim gerar espectros das estrelas para a calibração do SEA.

Com este trabalho contribui-se na área de projetos e simulações ópticas para uso espacial utilizando conceitos de IA e métodos clássicos de otimização. Abaixo são elencadas as principais contribuições deste trabalho:

- i. Estudo e implementação de dois algoritmos evolutivos propostos por pesquisadores do próprio INPE, o GEO e o M-GEO, que demonstraram ser muito versáteis como ferramentas para resolver problemas de engenharia.
- ii. Desenvolvimento de algoritmos e modelagem matemática para simulação de equipamentos ópticos.
- iii. Implementação de algoritmos clássicos de otimização, os algoritmos de Levenberg-Marquardt e do Gradiente Descendente, aplicados no controle da FLS de forma a conseguir simular espectros contínuos de estrelas.
- iv. Desenvolvimento de equipamento composto por hardware e software para simulação de estrelas necessárias para calibração do SEA.

1.2 Motivação

A Divisão de Eletrônica Aeroespacial do INPE (DEA) vem realizando diversos estudos para o desenvolvimento de um sensor de estrelas desde 1997. O objetivo principal é construir um sensor de estrelas autônomo que possa ser utilizado em missões espaciais. O desenvolvimento de sensores de estrelas requer o domínio de várias áreas do conhecimento, entre elas a óptica, fotônica, radiometria, mecânica de precisão e estrutural. Dentre os sensores de atitude, os de estrelas são os mais complexos, envolvendo diversas especialidades de conhecimento de forma atual e desafiadora para a tecnologia brasileira.

Estudos referentes a técnicas e algoritmos de identificação de estrelas iniciaram no INPE com os esforços de [Mejía Cabeza \(1997\)](#), que desenvolveu um sensor de estrelas utilizando uma câmera CCD comercial acoplada a uma objetiva do tipo telefoto. O sensor desenvolvido por Mejía Cabeza foi utilizado no sistema de determinação de

atitude do telescópio MASCO do INPE. [Carvalho \(2000\)](#) e [Fialho \(2007\)](#) também proporcionaram avanços na área de algoritmos de identificação de estrelas com o desenvolvimento de ambientes de simulação e testes para comparação dos diversos tipos de algoritmos que implementam técnicas para identificação de estrelas.

Estudos dos erros sistemáticos e aberrações ópticas inerentes aos sensores de estrelas foram inicialmente realizados por [Matos \(1997\)](#) e posteriormente aprofundados por [Albuquerque \(2005\)](#).

Em 2001 iniciou-se o desenvolvimento do SEA, e em paralelo também foram projetados e desenvolvidos diversos equipamentos de testes e calibração, que compõem a infraestrutura conhecida por GSE. Um dos itens mais importantes do GSE é o SE, que foi totalmente projetado e construído no INPE. O projeto e construção do SE apresentou diversos desafios que exigiram o desenvolvimento de uma metodologia utilizando várias áreas do conhecimento para sua superação, motivando o desenvolvimento deste trabalho.

1.3 Organização do Trabalho

O trabalho está estruturado da seguinte forma: no Capítulo 2 são apresentadas as fundamentações teóricas utilizadas no desenvolvimento da metodologia, expondo informações sobre o SEA e o SE, juntamente com os conceitos radiométricos e a explicação do algoritmo de Otimização Extrema Generalizada (GEO, do inglês *Generalized Extremal Optimization*) e sua versão multiobjetivo, o M-GEO (do inglês *Multiobjective GEO*).

No Capítulo 3 é apresentada a metodologia desenvolvida para definição de parâmetros do projeto da FLS. É demonstrado passo-a-passo a modelagem dos LEDs e o desenvolvimento das ferramentas computacionais usadas nesta tarefa. Ao final do capítulo é apresentado diversos testes para validar a escolha dos LEDs.

O Capítulo 4 apresenta o desenvolvimento do sistema de controle com retroalimentação capaz de controlar a FLS e assim gerar espectros de estrelas necessários para a calibração do SEA.

Finalmente, no Capítulo 5 são apresentadas as principais conclusões deste trabalho e sugestões para trabalhos futuros.

2 FUNDAMENTOS TEÓRICOS

Neste capítulo são apresentadas as bases teóricas que foram aplicadas no desenvolvimento das tarefas propostas neste trabalho. São descritas brevemente as partes e o funcionamento do SEA, a arquitetura do SE, as equações e conceitos necessários para a modelagem das estrelas e dos LEDs. Também são apresentados os algoritmos de otimização evolutiva GEO e M-GEO.

2.1 Sensor de Estrelas Autônomo (SEA)

Um sensor de estrelas é um equipamento eletro-óptico utilizado para determinar a atitude de veículos espaciais ou suborbitais utilizando como fonte de referência as estrelas (WERTZ, 1978; WERTZ; LARSON, 1999). O sensor é basicamente semelhante a uma câmera de vídeo digital, com objetiva óptica, matriz de detectores e eletrônica de processamento. As imagens adquiridas pelo sensor são processadas com o intuito de determinar as estrelas presentes no campo de visada (WERTZ; LARSON, 1999; BERTOLINO, 2004).

Os sensores de estrelas podem ser autônomos ou não. O sensor autônomo contém um processador de bordo responsável pela determinação da atitude do equipamento e um catálogo de estrelas armazenado em sua base de dados. A determinação de atitude corresponde a identificar a região do espaço para a qual o sensor está apontando, ou seja, relaciona de forma precisa as estrelas presentes no campo de visada com as que constituem o catálogo de estrelas (WERTZ; LARSON, 1999).

O sensor não autônomo apenas informa a magnitude e a posição angular das estrelas presentes em seu campo de visada. Neste caso, o processamento e a determinação de atitude fica a cargo do computador de bordo do veículo espacial ou da estação de controle em Terra.

Os sensores de estrelas possuem basicamente dois modos de operação: busca e rastreamento. No modo busca, as imagens adquiridas pelo sensor autônomo são processadas de forma a identificar as estrelas contidas no campo de visada, permitindo ao sensor a determinação da sua atitude. O sensor não autônomo no mesmo modo de operação apenas informa as estrelas presentes no campo de visada juntamente com suas respectivas magnitudes e coordenadas angulares.

No modo rastreamento, algumas estrelas mais brilhantes presentes no campo de visada são escolhidas para serem rastreadas pelo sensor. O tempo de processamento é reduzido, pois existe um conhecimento à priori da posição das estrelas com uma certa margem

de erro. Nesse mesmo modo, os sensores autônomos ou não autônomos tem seu funcionamento semelhante, pois se limitam a apenas informar a nova posição das estrelas rastreadas a cada aquisição (WERTZ; LARSON, 1999).

Considerado estado da arte em sensores de atitude, os sensores de estrelas destacam-se por sua precisão, baixo consumo de energia, ausência de partes móveis (o que aumenta sua confiabilidade e vida útil), versatilidade (podendo assumir diferentes modos de operação), tamanho e peso bem reduzidos e estabilidade por serem isentos de irregularidades nas medidas (como aquelas presentes no campo magnético da Terra que influenciam os magnetômetros e as derivas que são inerentes aos giroscópios) entre outros (BERTOLINO, 2004; ALBUQUERQUE, 2005; FIALHO, 2007).

Segundo Fialho (2007), a eletrônica do SEA em desenvolvimento na DEA é baseada em circuitos integrados digitais utilizando FPGA (do inglês *Field Programmable Gate Arrays*) e um processador de arquitetura SPARC V7, que possibilita a utilização de algoritmos de reconhecimento de padrão e a determinação da atitude do veículo espacial, caracterizando o sensor de estrelas como autônomo. Para imageamento, o SEA utiliza um detetor do tipo APS (do inglês *Active Pixel Sensor*), que opera na faixa espectral de 360 a 1 000 nm, e permite ao SEA identificar estrelas de magnitude visual zero a cinco.

O sensor APS apresenta muitas vantagens em relação ao detetor do tipo CCD utilizado em sensores de estrelas mais antigos. Diferente do CCD, o detetor APS permite a leitura de pequenos subconjuntos da matriz de pixels, possibilitando a execução do modo de operação rastreo de forma muito mais simples e eficiente. Possui matriz fotodetectora de 1024 x 1024 pixels, e incorpora conversor analógico/digital no próprio circuito integrado do detetor, além de usar apenas uma fonte de alimentação de 5V, com um consumo muito menor em relação ao detetor CCD, que ainda precisa de vários níveis de fontes de alimentação (BERTOLINO, 2004; FIALHO, 2007).

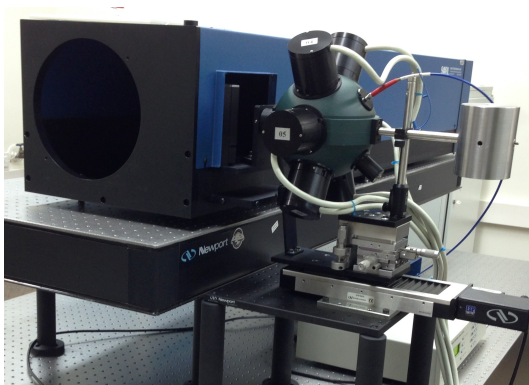
O detetor APS utilizado no SEA combina várias funções de uma câmera em um único circuito integrado, da captura da radiação eletromagnética à saída de vídeo digital, permitindo reduzir consideravelmente a quantidade de componentes da eletrônica digital, melhorando a confiabilidade, facilitando a miniaturização e possibilitando a programação de vários parâmetros no próprio sensor (FIALHO, 2007). A Figura 2.1 apresenta o Sensor de Estrelas Autônomo em desenvolvimento no INPE.



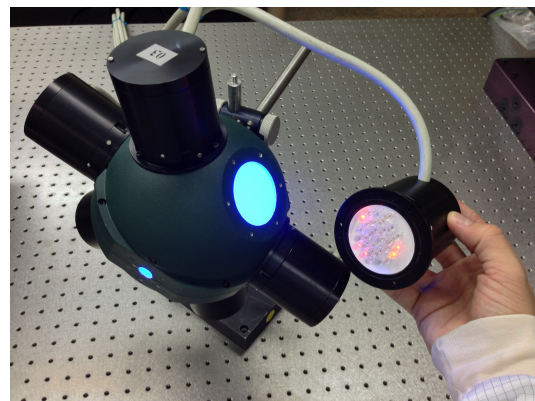
Figura 2.1 - Sensor de Estrelas Autônomo em desenvolvimento no INPE.

2.2 Simulador de Estrelas (SE)

Para caracterização, testes e calibração do SEA é necessário uma infraestrutura de testes sofisticada, o GSE. Dentre os diversos itens que compõem o GSE, destaca-se o SE, que é o responsável em gerar espectros eletromagnéticos semelhantes aos de estrelas. A Figura 2.2(a) apresenta o Simulador de Estrelas, e a Figura 2.2(b) a esfera integradora com as cabeças de fixação dos LEDs. Assim como o SEA, o SE também foi projetado e construído pela Divisão de Eletrônica Aeroespacial do INPE.



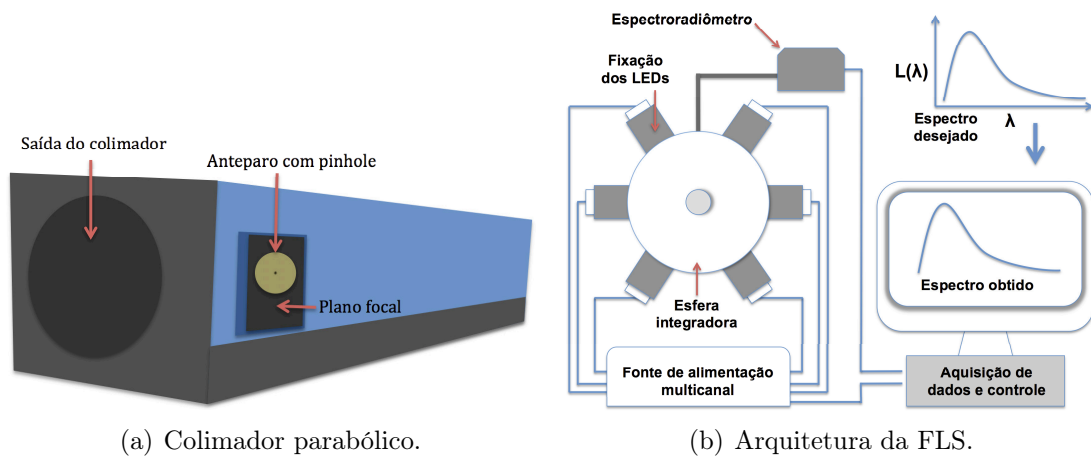
(a) Simulador de Estrelas.



(b) Esfera Integradora com LEDs.

Figura 2.2 - Configuração do Simulador de Estrelas construído no INPE.

O SE é formado por uma FLS, um colimador parabólico fora de eixo e um *pinhole* em seu plano focal. A FLS é o equipamento principal do SE, e é responsável por gerar espectros contínuos semelhantes aos emitidos por estrelas. O *pinhole* é o orifício por onde os feixes de luz gerados pela fonte de luz incidem e são projetados para dentro do colimador. O colimador, por sua vez, tem a função de tornar paralelo o feixe luminoso divergente recebido através do *pinhole* simulando uma fonte de luz no infinito, ou seja, irradiação espectral semelhante ao de uma estrela. A Figura 2.3(a) apresenta os principais itens do colimador, e a Figura 2.3(b) a arquitetura da FLS.



(a) Colimador parabólico.

(b) Arquitetura da FLS.

Figura 2.3 - Principais itens que compõem o Simulador de Estrelas.

A FLS é uma fonte de luz formada por uma esfera integradora iluminada por um grande número de LEDs com diferentes distribuições e picos espectrais. A esfera integradora é uma esfera oca, com pintura interna branca altamente reflexiva. A função da esfera integradora é gerar em sua saída a combinação dos espectros produzidos pelos diversos LEDs ligados em diferentes potências. Os LEDs podem ser controlados individualmente por meio de uma fonte de alimentação com 96 canais. Um espectrorradiômetro é utilizado para monitorar em tempo real a saída radiométrica e fotométrica da FLS. Os dados gerados pelo espectrorradiômetro são coletados por um computador que controla individualmente a corrente dos LEDs para obter a distribuição de potência espectral (DPE) desejada.

2.3 Modelagem da irradiação espectral das estrelas

Nesta seção é apresentada a teoria e as equações necessárias para modelar o espectro contínuo das estrelas. É apresentado o conceito de magnitude, seguido da modelagem da irradiação espectral das estrelas e da irradiância espectral desejada na saída do colimador.

2.3.1 Magnitude das estrelas

O SEA foi projetado para identificar estrelas na faixa de magnitude visual zero a cinco (BERTOLINO, 2004; ALBUQUERQUE, 2005; FIALHO, 2007). As magnitudes de estrelas são quantidades comparativas representadas por uma escala logarítmica, e estão relacionadas inversamente com o brilho das estrelas. Quanto maior a magnitude visual de uma estrela, menor é o seu brilho. A magnitude visual é definida pela seguinte equação (ALLEN, 1973):

$$m_{v1} - m_{v2} = 2,5 \log \left(\frac{I_2}{I_1} \right), \quad (2.1)$$

onde m_{v1} e m_{v2} são as magnitudes visuais de duas estrelas diferentes, e I_1 e I_2 suas respectivas irradiâncias espectrais.

Por ser uma quantidade comparativa, para calcular a magnitude visual é necessário uma referência. Allen (1973) define que a magnitude visual de uma estrela é igual a zero quando sua irradiância fora da atmosfera terrestre for $4 \times 10^{-9} \text{ erg cm}^{-2} \text{ s}^{-1} \text{ \AA}^{-1}$ em 550 nm, ou seja, equivalente a $4 \times 10^{-15} \text{ W cm}^{-2} \text{ nm}^{-1}$. Sendo assim, tornando m_{v1} a magnitude zero, e $I_1 = 4 \times 10^{-15}$, é possível calcular o valor da irradiância de qualquer estrela, para o comprimento de onda de 550 nm, em função da sua magnitude reescrevendo a Equação 2.1:

$$I_{550} = 10^{-0,4 m_v} 4 \times 10^{-15}, \quad (2.2)$$

onde I_{550} é a irradiância da estrela para o comprimento de onda de 550 nm, e m_v a magnitude visual da estrela.

2.3.2 Irradiância espectral das estrelas

A irradiância espectral de uma estrela pode ser modelada por um corpo negro (ALLEN, 1973; HALLIDAY et al., 1996). Um corpo negro é um objeto que absorve toda a radiação eletromagnética que incide sobre ele. Nenhuma radiação passa ou é refletida através dele (ALLEN, 1973; HALLIDAY et al., 1996; SMITH, 1996; SMITH, 2007; SCHUBERT, 2006). Um corpo negro ideal, em equilíbrio térmico, irradia energia na mesma taxa que absorve, e seu espectro de radiação é caracterizado por um único parâmetro T , a temperatura do corpo negro. A exitância espectral do corpo negro foi derivada pela primeira vez por Max Planck (1900) (HALLIDAY et al., 1996; SCHUBERT, 2006), e é dada por:

$$M(\lambda) = \frac{2hc^2}{\lambda^5 \left[\exp\left(\frac{hc}{\lambda kT}\right) - 1 \right]}. \quad (2.3)$$

A unidade de $M(\lambda)$ é $\text{W m}^{-2} \text{ nm}^{-1}$, ou seja, é a energia emitida por unidade de superfície de área (m^2) por unidade de comprimento de onda λ (nanometro). T é a temperatura do corpo negro em Kelvin (K), k é a constante de Boltzmann ($1,3806488 \times 10^{-23} \text{ J/K}$), h é a constante de Planck ($6,62606957 \times 10^{-34} \text{ J}\cdot\text{s}$) e c é a velocidade da luz ($299\,792\,458 \text{ m/s}$).

Para modelar uma estrela é necessário então saber sua temperatura e sua magnitude. Para obter a irradiância espectral da estrela a ser simulada, primeiramente é calculada a exitância espectral do corpo negro utilizando a Equação 2.3. Em seguida é calculado o valor da irradiância da estrela para o comprimento de onda de 550 nm utilizando a Equação 2.2. Após isso, é necessário calcular a seguinte razão:

$$R_{550} = \frac{I_{550}}{M(550)}, \quad (2.4)$$

onde R_{550} é a razão entre o valor da irradiância da estrela, I_{550} , e a exitância do corpo negro, $M(550)$, no comprimento de onda de 550 nm. Dessa forma, a irradiância espectral da estrela é calculada pela seguinte equação:

$$I(\lambda) = M(\lambda) R_{550}. \quad (2.5)$$

A Figura 2.4 apresenta a irradiância espectral fora da atmosfera terrestre de uma estrela de magnitude zero e temperatura de 7500K modelada utilizando o passo-a-passo descrito.

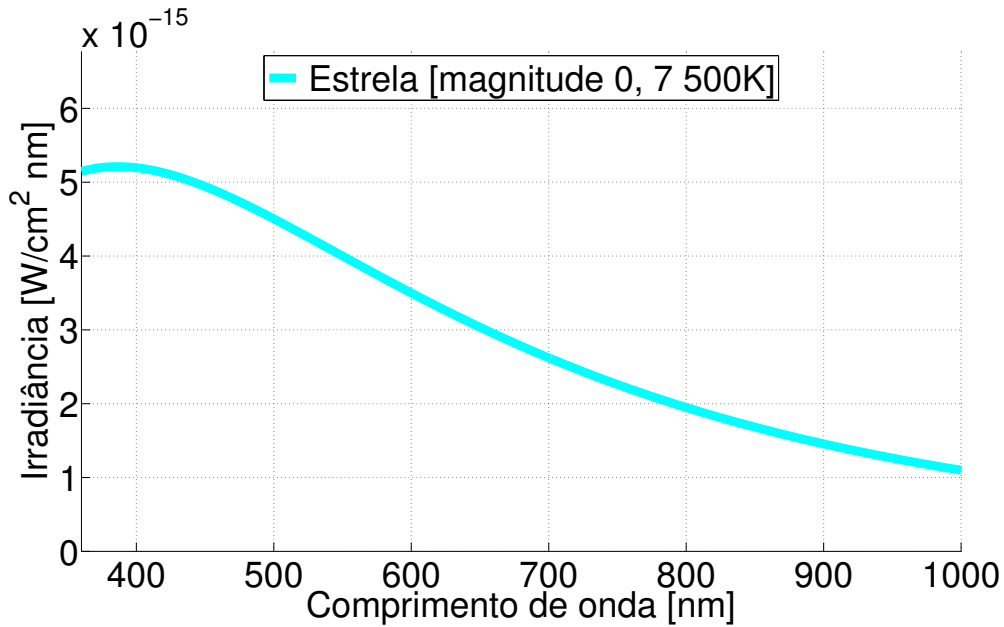


Figura 2.4 - Irradiância espectral de uma estrela de magnitude zero a 7500K.

A característica espectral da fonte de luz é muito importante para simular estrelas de maneira adequada. Estudos a cerca das características espectrais das estrelas no intervalo de magnitude que o SEA opera foram conduzidos por Albuquerque (2005). Para essa tarefa utilizou-se o Catálogo de Estrelas Hipparcos disponibilizado pelas agências European Space Agency (ESA) (1997) e Centre de Donnés Astronomiques de Strasbourg (CDS) (2007), selecionando-se apenas as estrelas na faixa de magnitude zero a cinco, num total de 1625 estrelas. Em virtude do catálogo não apresentar informações sobre as temperaturas das estrelas, calculou-se a temperatura efetiva aproximada de cada estrela utilizando as equações abaixo (REED, 1998):

$$T = 10^{(3,402 - \sqrt{0,515 + 1,376(B-V)}) / 0,688} \quad \text{para } -0,375 < (B - V) < -0,041, \quad (2.6a)$$

$$T = 10^{((B-V) - 14,551) / -3,684} \quad \text{para } (B - V) \geq -0,0413, \quad (2.6b)$$

onde T é a temperatura em Kelvin, (B) e (V) são filtros especiais utilizados na composição do índice de cor Johnson. Cada filtro está posicionado em uma banda específica do espectro eletromagnético: (B) centrado em 445 nm (região do azul) e (V) centrado em 551 nm (região do visível) (EUROPEAN SPACE AGENCY (ESA), 1997).

A Figura 2.5 apresenta o histograma com a distribuição das temperaturas das estrelas na faixa de magnitude zero a cinco. Por meio da análise estatística percebeu-se que a maioria das estrelas a serem simuladas estão na faixa de temperatura entre 3 000K e 20 000K.

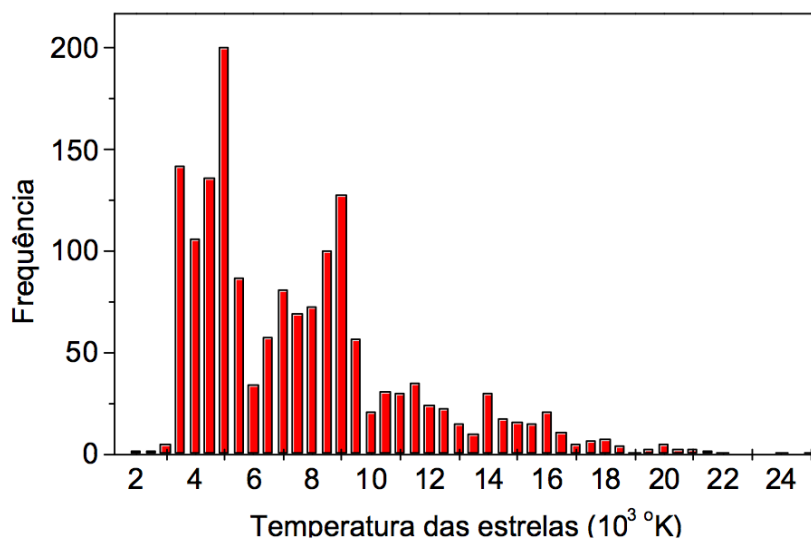


Figura 2.5 - Histograma de temperatura das estrelas de magnitudes zero a cinco, selecionadas do Catálogo de Estrelas Hipparcos.
 Fonte: Albuquerque (2005).

2.3.3 Irradiância espectral desejada na saída do colimador

Para simular uma estrela com temperatura e magnitude predeterminada, a irradiância espectral dessa estrela deve ser reproduzida na saída do colimador. Durante o processo de simulação a esfera integradora gera em sua saída uma radiância espectral proveniente da combinação dos espectros produzidos pelos diversos LEDs ligados em diferentes potências. Essa radiância espectral é projetada para dentro do colimador através do *pinhole*. O feixe radiante divergente recebido no colimador é então rearranjado de forma paralela e projetado em sua saída como a irradiância espectral da estrela simulada.

A Figura 2.6 apresenta os detalhes de como a esfera integradora é usada juntamente com o pinhole e colimador para simular o espectro contínuo de estrelas.

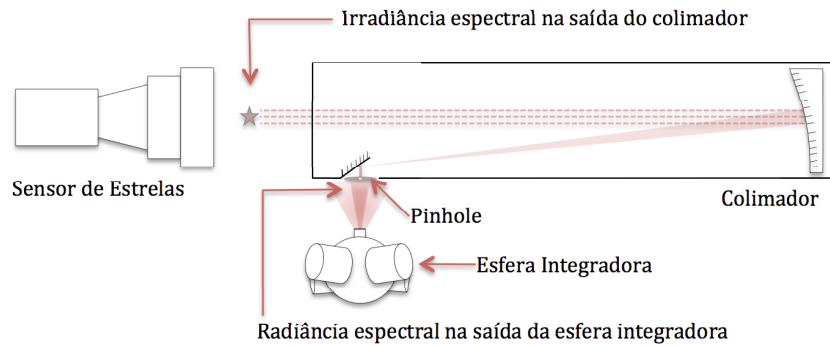


Figura 2.6 - Simulação da irradiância espectral de uma estrela no Simulador de Estrelas.

A radiância espectral é a energia emitida ou transmitida por unidade de área, ângulo sólido e comprimento de onda em uma dada direção ($\text{W}/\text{cm}^2 \text{ sr nm}$) (OPTICAL SOCIETY OF AMERICA, 1978; SMITH, 1996; SMITH, 2007). A radiância é definida por:

$$L = \frac{\partial^2 \Phi}{\partial \omega (\cos \theta \partial A)}, \quad (2.7)$$

onde Φ é o fluxo, dA é a área elementar contendo um determinado ponto, $d\omega$ é o ângulo sólido numa determinada direção e θ o ângulo entre a reta normal a área e uma dada direção conforme pode ser observado na Figura 2.7.

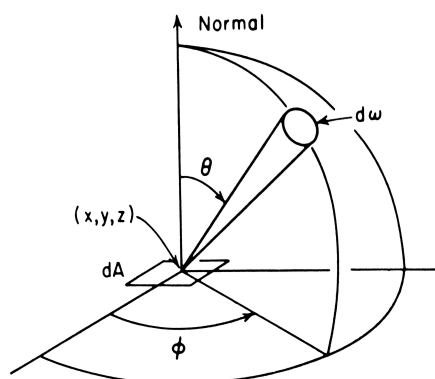


Figura 2.7 - Energia emitida por unidade de área e ângulo sólido em uma dada direção.
Fonte: OPTICAL SOCIETY OF AMERICA (1978).

A irradiância espectral é a energia recebida ou transmitida por unidade de área e comprimento de onda ($\text{W}/\text{cm}^2 \text{ nm}$) (OPTICAL SOCIETY OF AMERICA, 1978; SMITH, 1996; SMITH, 2007). A irradiância é definida por:

$$E = \frac{\partial\Phi}{\partial A}. \quad (2.8)$$

Para estimar a irradiância espectral na saída do colimador a partir do fluxo espectral produzido na esfera integradora, foi utilizado a equação abaixo (LABSPHERE, 2012):

$$L = \left(\frac{\Phi_e}{\pi A_s} \right) \left(\frac{\rho}{(1 - \rho) \left(1 - \left(\frac{A_i + A_e}{A_s} \right) \right)} \right). \quad (2.9)$$

A Equação 2.9 fornece a radiância espectral L ($\text{W}/\text{cm}^2 \text{ sr nm}$) na saída da esfera integradora para um dado fluxo radiante Φ_e produzido pelos LEDs. A_s é área interna da esfera integradora, A_i a área da porta de entrada da esfera integradora, A_e a área da porta de saída da esfera integradora e ρ o fator de reflectância interna da esfera.

Com a equação abaixo é possível calcular a irradiância espectral na saída do colimador dada uma radiância espectral em seu plano focal (LABSPHERE, 2012):

$$E = 2 \pi L A_p \frac{1}{8 (f/\#)^2 A_c}, \quad (2.10)$$

onde A_p é a área do *pinhole*, A_c a área do feixe do colimador (porta de entrada da esfera integradora) e $f/\#$ é a razão entre a distância focal e o diâmetro da porta de entrada do colimador.

A FLS poderia ser desenvolvida utilizando lâmpadas de filamento (WALL et al., 2001), todavia seria muito difícil simular estrelas de temperaturas acima de 3 500K. Esse tipo de fonte de luz apresenta baixo fluxo devido a DPE da lâmpada, em especial na região do ultravioleta e azul (FRYC et al., 2005a).

Se fossem usadas lâmpadas de filamentos na construção da FLS, para possibilitar a simulação de estrelas de temperaturas mais elevadas, uma das alternativas seria utilizar filtros espectrais projetados especificamente para esta aplicação. O problema com esta solução estaria na baixa eficiência luminosa e na dissipação de calor da lâmpada de filamento que poderia deformar o colimador, modificando assim a posição

relativa da estrela durante os testes. Devido a esses fatores, foi decidido desenvolver uma fonte de luz espectralmente sintonizável baseada em LEDs.

2.4 Teoria e equações para modelagem dos LEDs

Utilizando LEDs com diferentes DPEs e controlando o fluxo radiométrico de saída de cada tipo de LED, é possível produzir uma fonte de luz espectralmente sintonizável capaz de gerar espectros semelhantes ao de um alvo específico com alta acurácia (FRYC et al., 2005b). Todavia a qualidade da simulação dos espectros desejados está intimamente ligada a escolha adequada dos LEDs.

A escolha ótima da quantidade e tipos de LEDs necessários para que a FLS consiga simular estrelas de forma a atender aos requisitos do SEA é um problema multiobjetivo, onde procura-se minimizar a quantidade de LEDs e ao mesmo tempo maximizar a performance da FLS em simular estrelas.

Levando-se em consideração a faixa espectral de operação do SEA, foram selecionados e modelados 160 tipos de LEDs cobrindo a faixa espectral de 360 a 1000 nm. A dificuldade neste processo foi obter o fluxo espectral de cada LED, pois os *datasheets* não trazem essa informação diretamente, e a informação de potência é fornecida em diferentes unidades, tais como candela, watt, watt/esferorradiano e lumens.

Usando as informações técnicas disponíveis nos *datasheets*, tais como diagrama de radiação, diagrama de intensidade luminosa relativa, ângulo de visão, largura de meia potência ou FWHM (do inglês *Full Width at Half Maximum*), potência radiante e intensidade radiante, foi estimada a potência espectral de cada um dos LEDs candidatos a serem usados na FLS.

2.4.1 Distribuição de potência espectral de um LED

A DPE dos LEDs foi modelada baseando-se em modelos matemáticos apresentados em Ohno (2004), Ohno (2005), Schubert (2006) e Commission Internationale de l'Eclairage (CIE) (2007). A DPE de um LED, $S_{LED}(\lambda)$, para um comprimento de onda de pico λ_0 e largura de banda (comprimento da metade da potência espectral) $\Delta\lambda_{0,5}$ é dada por

$$S_{LED}(\lambda, \lambda_0, \Delta\lambda_{0,5}) = \frac{g(\lambda, \lambda_0, \Delta\lambda_{0,5}) + 2g^5(\lambda, \lambda_0, \Delta\lambda_{0,5})}{3}, \quad (2.11)$$

onde $g(\lambda, \lambda_0, \Delta\lambda_{0,5}) = \exp(-[(\lambda - \lambda_0)/\Delta\lambda_{0,5}]^2)$. O comprimento de onda é dado

em nanômetros. Na Figura 2.8 é apresentado o exemplo das DPEs de dois LEDs modelados e comparados com as DPEs reais, medidas no INPE utilizando um espectrorradiômetro.

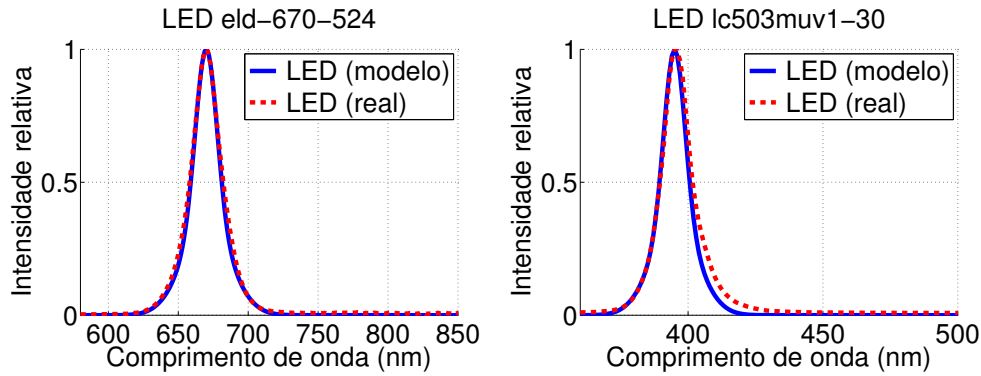


Figura 2.8 - DPE de dois LEDs modelados comparados com LEDs reais.

2.4.2 Distribuição de radiação de um LED por ângulo

A intensidade de radiação de um LED varia de acordo com o ângulo de observação, e em geral, essa informação não é fornecida pelo fabricante. Para os casos em que essa informação é necessária e não consta na documentação do LED, ou até mesmo na falta de equipamentos próprios para realizar a medição, a distribuição aproximada da radiação por ângulo de um LED pode ser modelada por uma Super Gaussiana (RADIANT ZEMAX LLC, 2011) dada por:

$$I(\theta_x, \theta_y) = I_0 e^{-2((\theta_x/\alpha_x)^{2G_x} + (\theta_y/\alpha_y)^{2G_y})}, \quad (2.12)$$

onde α_x é o ângulo divergente entre XZ em graus, G_x é o fator da Super Gaussiana para a direção X, com definição similar para os valores de Y. Note que se G_x e G_y forem iguais a 1,0 resulta em uma distribuição Gaussiana típica. Os valores de G_x e G_y devem estar entre 0,01 e 50,0 (RADIANT ZEMAX LLC, 2011).

Para radiações com perfil de distribuição gaussiano, a largura de banda espectral é definida como o comprimento da faixa de radiação onde a potência de emissão é a metade da potência máxima, como pode ser observado na Figura 2.9. Na literatura esse comprimento também é referenciado como FWHM. O ângulo formado entre os pontos onde a potência de emissão do LED é igual a metade da potência total é referenciado como θ_{fwhm} ou $2\theta_{\frac{1}{2}}$ (duas vezes o ângulo de meia potência) como exemplificam as Figuras 2.9 e 2.10.

Para uma distribuição Gaussiana típica, definindo a equação para $\frac{1}{2} I_0$, θ_y igual a zero, substituindo o valor de θ_x por $\frac{1}{2}\theta_{fwhm}$, então define-se α_x dado por

$$\alpha_x = \frac{\theta_{fwhm}}{\sqrt{2 \ln(2)}} \quad \text{ou} \quad \alpha_x = (0,8493218) \theta_{fwhm}. \quad (2.13)$$

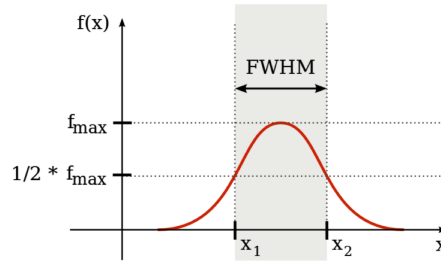


Figura 2.9 - FWHM da distribuição de radiação com perfil gaussiano.
Fonte: Adaptado de Schubert (2006).

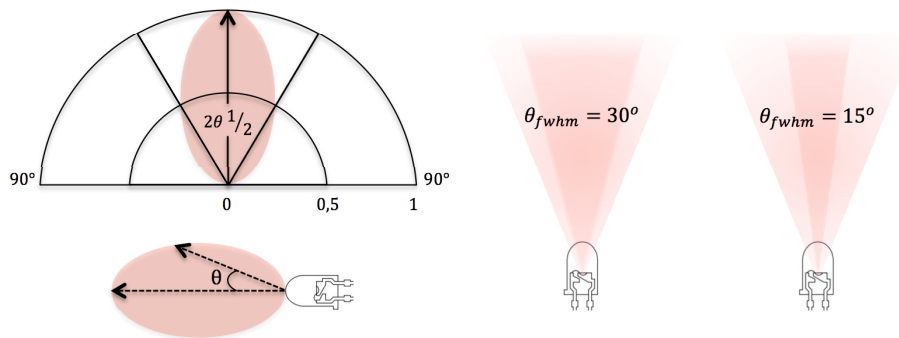


Figura 2.10 - Ângulo de meia potência (θ_{fwhm} ou $2\theta_{\frac{1}{2}}$).
Fonte: Adaptado de Schubert (2006).

2.4.3 Cálculo do fluxo luminoso de um LED

Se o padrão de radiação é conhecido em termos da intensidade luminosa, o fluxo luminoso Φ_v pode ser calculado pela seguinte equação:

$$\Phi_v = \int_0^{2\pi} \int_0^\pi I_v(\theta, \phi) \sin \theta \, d\theta \, d\phi \quad (\text{lm}), \quad (2.14)$$

onde $I_v(\theta, \phi)$ é a intensidade luminosa em *candela*, e também obtida em função de um ângulo polar θ e um ângulo azimutal ϕ . Os ângulos são sempre em radianos.

Quando um padrão de radiação de uma fonte de luz é simétrico em torno do feixe principal (simétrico no ângulo azimutal ϕ), um único conjunto de medidas ao longo do ângulo polar θ é suficiente para descrever esse padrão. Para padrões de radiação simétricos em ϕ , a Equação 2.14 pode ser simplificada da seguinte forma:

$$\Phi_v = 2\pi \int_0^\pi I_v(\theta) \sin \theta d\theta \quad (\text{lm}). \quad (2.15)$$

Além das Equações 2.14 e 2.15, o fluxo luminoso também pode ser calculado utilizando a seguinte equação (COMMISSION INTERNATIONALE DE L'ECLAIRAGE (CIE), 2007; OHNO, 1997):

$$\Phi_v = K_m \int_\lambda \Phi_e(\lambda) V(\lambda) d\lambda \quad (\text{lm}), \quad (2.16)$$

onde K_m é uma constante com valor igual a 683 lumens/W em 555 nm. $\Phi_e(\lambda)$ é o fluxo radiante por comprimento de onda (W/nm), e $V(\lambda)$ é a resposta espectral do olho humano.

Tipicamente os valores $I_v(\theta)$ são dados como um conjunto de medições discretas. O método para integração numérica apresentado abaixo se aproxima da integral definida na Equação 2.15 usando a regra do trapézio.

$$\Phi_v \approx 2\pi \sum_{i=1}^{N-1} \left[\frac{f_i + f_{i-1}}{2} \Delta_i \right] \quad (\text{lm}). \quad (2.17)$$

A Equação 2.17 apresenta o método numérico para calcular o valor aproximado do fluxo luminoso Φ_v conhecendo-se o valor da intensidade luminosa I_v para cada ângulo θ_i . As funções intermediárias f_i e Δ_i são calculadas da seguinte forma:

$$f_i = F_i \sin \left(\frac{\pi \theta_i}{180} \right), \quad (2.18)$$

e

$$\Delta_i = \frac{\pi}{180} (\theta_i - \theta_{i-1}), \quad (2.19)$$

onde F_i é o padrão de radiação de entrada por ângulo $I_v(\theta_i)$ para a intensidade luminosa. O termo $\pi/180$ que multiplica o valor do ângulo é necessário para converter o ângulo em graus para radianos.

Outra situação comum surge nos casos onde o padrão de radiação é apresentado de forma normalizada ou relativa. Nesses casos, a forma do padrão de radiação determina um ângulo sólido. Em geral, o ângulo sólido para qualquer padrão de radiação pode ser equacionado da seguinte forma:

$$\Omega = \int_0^{2\pi} \int_0^\pi F_{norm}(\theta, \phi) \text{sen } \theta \, d\theta \, d\phi \quad (\text{sr}). \quad (2.20)$$

Para padrões simétricos de radiação em ϕ , a Equação 2.20 pode ser simplificada para:

$$\Omega = 2\pi \int_0^\pi F_{norm}(\theta) \text{sen } \theta \, d\theta \quad (\text{sr}). \quad (2.21)$$

A equação abaixo apresenta o método numérico utilizado para padrões de radiação normalizado ou relativo. Assim, a Equação 2.21 para o cálculo do ângulo sólido pode ser aproximada por:

$$\Omega \approx 2\pi \sum_{i=1}^{N-1} \left[\frac{f_i + f_{i-1}}{2} \Delta_i \right] \quad (\text{sr}), \quad (2.22)$$

onde as funções intermediárias f_i e Δ_i são calculadas da seguinte forma:

$$f_i = F_i \text{sin} \left(\frac{\pi \theta_i}{180} \right), \quad (2.23)$$

e

$$\Delta_i = \frac{\pi}{180} (\theta_i - \theta_{i-1}). \quad (2.24)$$

A Equação 2.17 é semelhante a Equação 2.22, todavia com a Equação 2.22 calcula-se o ângulo sólido de um padrão de radiação enquanto que a Equação 2.17 é utilizada para calcular o fluxo luminoso.

2.5 O Problema de Otimização

Problemas de otimização em engenharia normalmente são de natureza multiobjetivo, ou seja, com objetivos ou critérios conflitantes (VANDERPLAATS, 1998; EIBEN; SMITH, 2007; BRANKE et al., 2008). A menos que haja uma função que defina a relação de importância entre os objetivos, eles devem ser tratados como igualmente independentes. A maneira mais simples de se tratar diversos objetivos é o de transformar a função de adaptação em uma soma ponderada dos vários objetivos $f = [f_1, \dots, f_k]$:

$$g(f) = \sum_{i=1}^k W_i f_i \quad (2.25)$$

O conhecimento dos pesos W_i introduz, porém, uma nova questão: como quantificar a importância de cada objetivo em relação aos outros. O tratamento de alguns dos objetivos com restrições também é feito, todavia um dos perigos desta técnica é a de se restringir em excesso, ou de modo insuficiente, o espaço de soluções.

Quando um problema é tratado como multiobjetivo, não é encontrado apenas uma solução, mas sim um conjunto de soluções. Uma solução desse conjunto não pode ser considerada, a princípio, melhor do que uma outra solução do mesmo conjunto, isso devido ao fato de que essa solução será pior do que outra solução pelo menos num determinado objetivo (EIBEN; SMITH, 2007; BRANKE et al., 2008). Este conjunto de soluções é conhecido como soluções não dominadas. Quando essas soluções são plotadas no espaço de funções objetivo, elas formam a assim chamada fronteira de Pareto.

Para avaliar as características de uma solução são utilizadas métricas chamadas funções objetivo. O exemplo considerado na Figura 2.11 representa um problema de minimização com duas funções objetivo, ou seja, quanto menor os valores de F1 e F2, melhor é a solução. Os círculos representam soluções não dominadas e os triângulos representam soluções dominadas.

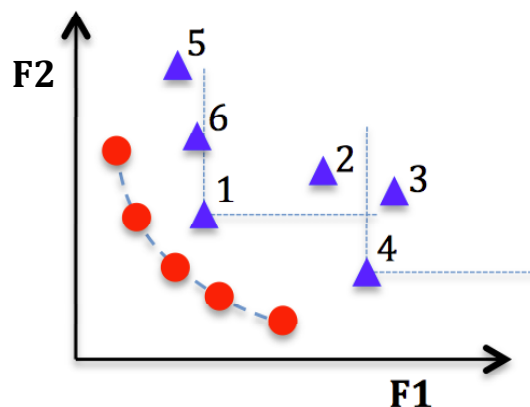


Figura 2.11 - Soluções para um problema multiobjetivo de minimização, plotado no espaço de funções objetivo F1 e F2.

Uma solução S_1 é dita dominar uma solução S_2 , se ela não for pior que a solução S_2 em nenhum dos objetivos, e deve ser estritamente melhor que S_2 em pelo menos um objetivo. A relação de dominância entre soluções dominadas também pode ser considerada; por exemplo, a solução 1 domina as soluções 2 e 3 (EIBEN; SMITH, 2007; BRANKE et al., 2008).

A meta em um problema multiobjetivo é encontrar soluções não dominadas que formam a fronteira de Pareto. Assim, após o processo de otimização multiobjetivo, um conjunto de soluções de compromisso são obtidas, e não somente uma única solução como acontece em uma abordagem mono-objetivo. Isso permite aos projetistas uma idéia mais clara dos compromissos que podem ser feitos no projeto considerando as métricas utilizadas.

Em problemas de otimização, há grande possibilidade de que o algoritmo aplicado na busca por soluções ótimas fique preso em um mínimo local, e para evitar isso foram desenvolvidos vários métodos que realizam uma busca global pela solução ótima no espaço de projeto (EIBEN; SMITH, 2007; BRANKE et al., 2008).

Uma classe desses métodos foram construídas inspiradas em processos que ocorrem na natureza, ou seja, tratam os problemas de otimização através de mecanismos numéricos que tentam mimetizar fenômenos naturais. Esses métodos fazem parte dos paradigmas de Inteligência Artificial, e são muito utilizados para resolver problemas que não possuem soluções por métodos clássicos. Na área de otimização, o Algoritmo Genético (AG) e o Recozimento Simulado (RS) são os que mais se destacam por virem sendo desenvolvidos e aplicados em um amplo número de problemas, tanto em ciência quanto em engenharia.

Recentemente, um novo método de busca global inspirado na natureza foi proposto por De Sousa et al. (2003), o algoritmo Otimização Extrema Generalizada (GEO, do inglês *Generalized Extremal Optimization*). Este método foi melhorado por Galski (2006), que propôs o M-GEO (uma versão multiobjetivo do GEO), e por Lopes et al. (2008), que propôs o GEO_{real} (versão do GEO com codificação real).

Os algoritmos de otimização evolucionária são ferramentas excelentes para resolver problemas de otimização multiobjetivo diretamente. Neste trabalho fez-se o uso da versão multiobjetivo do algoritmo Otimização Extrema Generalizada (M-GEO). Abaixo é descrito em mais detalhes o algoritmo GEO e sua versão multiobjetivo, o M-GEO.

2.5.1 Otimização Extrema Generalizada (GEO)

O GEO foi apresentado por De Sousa (2002) como uma generalização do algoritmo Otimização Extrema (EO) proposto por Boettcher e Percus (2001). Ambos são baseados num modelo simplificado de evolução natural de espécies apresentado por Bak e Sneppen (1993). No GEO, a codificação das variáveis de projeto é binária, assim como a codificação usada no Algoritmo Genético Simples (AGS). Todavia a string de bits não representa o cromossomo de uma solução candidata, mas diferentes espécies coexistindo em um ecossistema conforme pode ser observado na Figura 2.12.

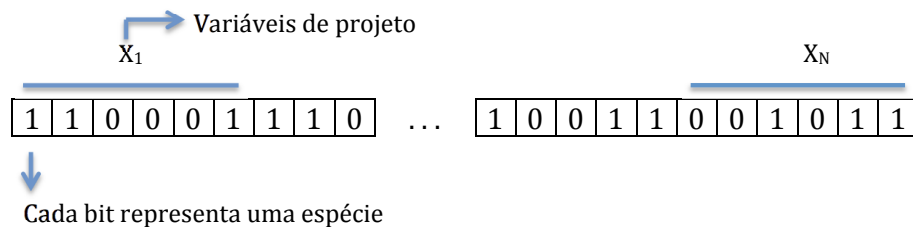


Figura 2.12 - Variáveis de projeto codificadas em uma string binária; neste exemplo cada variável é representada por seis bits.
 Fonte: Adaptado de De Sousa (2002).

O GEO possui algumas vantagens sobre os outros algoritmos evolutivos: tem somente um parâmetro livre de ajuste, a sua implementação é muito simples e tem demonstrado um desempenho competitivo para funções de teste e em problemas reais (DE SOUSA, 2002; DE SOUSA et al., 2003; DE SOUSA et al., 2004; ABREU et al., 2005; GALSKI, 2006; SWITALSKI; SEREDYNSKI, 2008; SWITALSKI; SEREDYNSKI, 2011; SWITALSKI; SEREDYNSKI, 2012).

Uma das primeiras decisões a serem tomadas na implementação prática do algoritmo, é definir o número de bits que representará cada uma das X_i variáveis de projeto. Este pode ser definido pela precisão que se deseja para o valor de cada variável X_i . A equação a seguir pode ser utilizada para calcular o número mínimo de bits necessário para se obter uma determinada precisão e_i (DE SOUSA, 2002; GALSKI, 2006):

$$l_i = \log_2 \left(1 + \frac{(X_{max} i - X_{min} i)}{e_i} \right), \quad (2.26)$$

onde l_i é o número mínimo de bits para representar a variável de projeto X_i . $X_{min} i$ e $X_{max} i$ são respectivamente os valores mínimo e máximo para a variável X_i . Note-se que o valor calculado para l_i através da Equação 2.26 pode ser um número não inteiro, e nesse caso atribui-se à l_i o próximo número inteiro.

Na inicialização do GEO ocorre o preenchimento de uma sequência binária C de comprimento L , por meio de L sorteios de 0s e 1s, ambos com igual probabilidade de ocorrência. Esta seqüência binária define uma configuração ou solução candidata a ótimo global, representada no sistema numérico utilizado internamente pelo algoritmo GEO (sistema binário).

Em geral o sistema numérico utilizado para representar as variáveis de projeto não é o sistema binário, mas sim decimal. Nesses casos, a fim de calcular a função objetivo, $F(X)$, é necessário primeiro converter C em X , ou seja, converter cada variável de projeto, X_i , $i \in \{1, \dots, N\}$, de sua codificação binária para seu valor decimal. Essa conversão é feita conforme a equação a seguir (DE SOUSA, 2002; GALSKI, 2006):

$$X_i = X_{min} i + (X_{max} i - X_{min} i) \frac{\sum_{j=1}^{l_i} C_j 2^{j-1}}{2^{l_i} - 1}, \quad (2.27)$$

onde $X_{min} i$ e $X_{max} i$ são respectivamente os valores mínimo e máximo para a variável X_i no sistema decimal. C_j é o valor do j -ésimo bit usado na representação de X_i , e l_i é o número de bits para a variável X_i .

O algoritmo GEO é composto dos seguintes passos (adaptado de De Sousa (2002) e Galski (2006)):

1. Inicialize aleatoriamente uma sequência binária de comprimento L que codifica N variáveis de projeto. Sendo cada variável codificada em uma subsequência de comprimento l_i , $i \in \{1, \dots, N\} \mid \sum_{i=1}^N l_i = L$. Para uma configuração inicial de bits C , converta C em X e calcule o valor da função objetivo $F(X)$ e faça $X_{melhor} = X$ e $F(X_{melhor}) = F(X)$.
2. Em uma dada iteração, para cada bit $i \in \{1, \dots, L\}$ da sequência C faça:
 - a) Altere o valor do bit i (de 0 para 1 ou de 1 para 0), obtendo uma configuração de bits C_i . Converta C_i em X_i e calcule o valor da função objetivo $F(X_i)$.
 - b) Calcule e atribua ao bit i o índice de adaptabilidade $\Delta F(X_i) = F(X_i) - F(X_{melhor})$. Esse índice indica o ganho ou perda que se tem ao mudar o valor do bit comparado com o melhor valor da função objetivo encontrado até o momento.
 - c) Retorne o bit i ao seu valor original.
3. Ordene os bits $i \in \{1, \dots, L\}$ de acordo com os seus índices de adaptabilidade, de $k = 1$ para o menos adaptado à $k = L$ para o mais adaptado se for um problema de minimização, e o contrário para um problema de maximização. Em um problema de minimização, valores altos de $\Delta F(X_i)$ terão maior *rank*, enquanto para problemas de maximização ocorre o oposto. Se ocorrer de dois ou mais bits apresentarem o mesmo valor para o índice de adaptabilidade, então ordene-os aleatoriamente com distribuição uniforme.
4. Escolha com igual probabilidade um bit candidato i para sofrer mutação (mudar de 0 para 1 ou de 1 para 0). Gere um número aleatório RAN com distribuição uniforme no intervalo $[0, 1]$. Se $P(k) \approx k^{-\tau}$ for maior ou igual a RAN , o bit é modificado. Caso contrário, o processo é repetido até que um bit seja modificado.
5. Para o bit escolhido para sofrer mutação faça $C = C_i$ e calcule $F(X)$.
6. Se $F(X) < F(X_{melhor})$ então faça $F(X_{melhor}) = F(X)$ e $X_{melhor} = X$.
7. Repita os passos 2 à 6 até que um dado critério de parada seja satisfeito.
8. Retorne X_{melhor} e $F(X_{melhor})$.

A Figura 2.13 apresenta uma visão geral do GEO sob a forma de fluxograma, adaptado de De Sousa (2002) e Galski (2006). Na figura, $F(X)$ é a função objetivo, k é o valor da posição ocupada pelo bit no ordenamento efetuado a cada iteração.

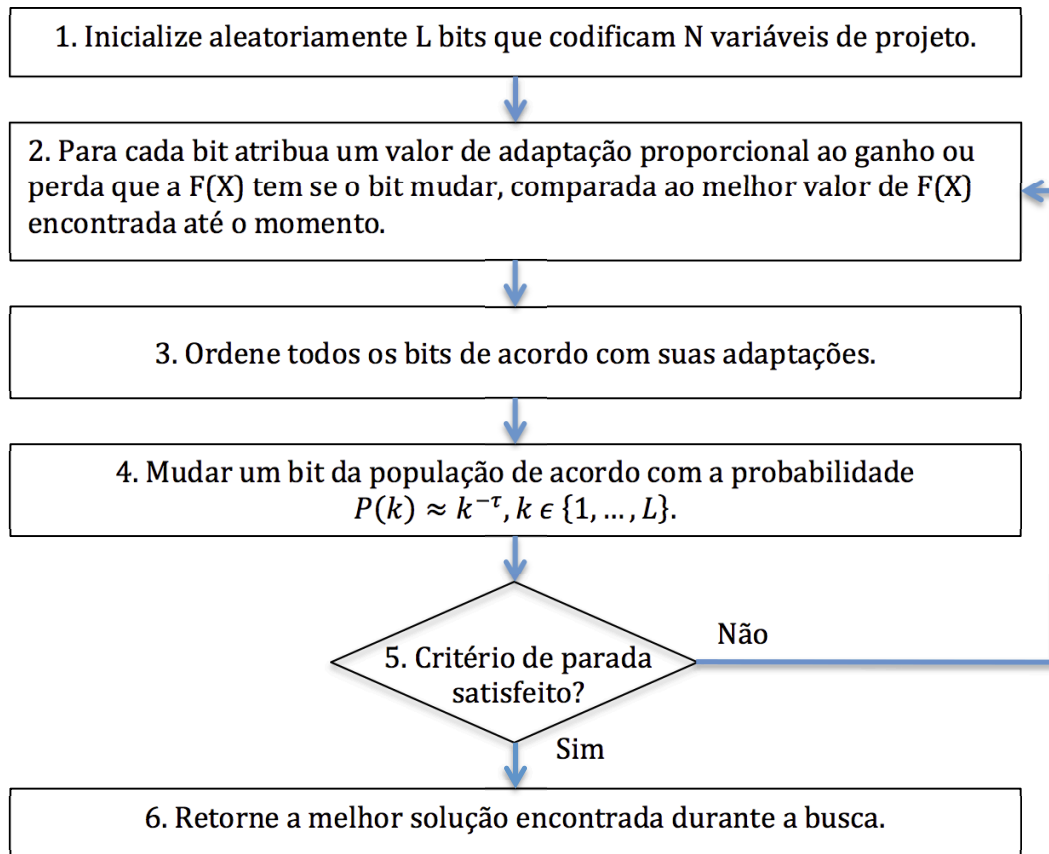


Figura 2.13 - Fluxograma do algoritmo GEO.

Fonte: Adaptado de De Sousa (2002) e Galski (2006).

2.5.2 Otimização Extrema Generalizada Multiobjetivo (M-GEO)

Nos últimos anos foram sugeridas diversas variações do GEO canônico também apresentando resultados competitivos, tanto para funções testes quanto para problemas reais (GALSKI, 2006; LOPES et al., 2008). O M-GEO foi proposto por Galski (2006) como a versão multiobjetivo do GEO.

A principal diferença entre otimização multiobjetivo e mono-objetivo é que ao invés de uma única função objetivo, passa-se a ter um vetor de funções objetivo com dimensão dada por $NFOBJ$, ou seja, o número de funções objetivo que se deseja minimizar ou maximizar simultaneamente.

Na verdade, o problema de otimização multiobjetivo pode ser visto como uma generalização do problema de otimização mono-objetivo. Um problema de otimização mono-objetivo é um caso particular onde $NFOBJ = 1$. Nesse caso, independentemente do número de ótimos globais existentes para o problema de otimização mono-objetivo, a fronteira de Pareto resultante terá apenas um único ponto em um espaço de funções objetivo unidimensional.

A principal alteração implementada na versão multiobjetivo do GEO ocorre no cálculo da adaptabilidade dos bits. Para cada geração no M-GEO, uma função objetivo dentre as $NFOBJ$ funções objetivo do problema é selecionada com distribuição uniforme e utilizada para conduzir a pesquisa. Outra mudança é que a cada geração as soluções são analisadas, e todas as soluções não dominadas são salvas, gerando o conjunto de Pareto, que é a saída gerada pelo algoritmo. O algoritmo M-GEO é composto dos seguintes passos (adaptado de Galski (2006)):

1. O primeiro passo do M-GEO é idêntico ao primeiro passo do GEO: preencha uma sequência binária C de L bits aleatoriamente com 0s and 1s.
2. Igualmente como é feito no GEO, para cada bit $i \in \{1, \dots, L\}$ da sequência C em uma dada iteração faça:
 - a) Altere o valor do bit i (de 0 para 1 ou de 1 para 0), obtendo uma configuração de bits C_i . Converta C_i em X_i e calcule o valor de todas as $NFOBJ$ funções objetivo, gerando um vetor função objetivo na forma $[F_1(X_i), F_2(X_i), \dots, F_{NFOBJ}(X_i)]$.
 - b) Crie o conjunto contendo a aproximação da fronteira de Pareto, verificando, alterando e salvando, conforme se fizer necessário.
 - c) Retorne o bit i ao seu valor original.
3. Selecione aleatoriamente uma função objetivo I_c entre todas as $NFOBJ$. Uma vez escolhida uma função objetivo I_c , a busca ocorre do quarto ao sétimo passo exatamente como acontece numa busca com o GEO.
4. Calcule e atribua para cada um dos L bits (e seus vetores X_i associados) o valor da função de adaptabilidade, baseado no ganho ou perda que a I_c -ésima função objetivo tem se o respectivo bit for comutado, comparado ao melhor valor encontrado até o momento para a I_c -ésima função objetivo. Matematicamente, o valor de adaptação é obtido calculando-se

$\Delta F = F_{Ic}(X_i) - \text{melhor } F_{Ic}$, $i \in \{1, \dots, L\}$, onde melhor F_{Ic} é o melhor valor de $F_{Ic}(X)$ encontrado durante a busca até a iteração anterior. (passos 2. até 7.).

5. Ordene os bits $i \in \{1, \dots, L\}$ de acordo com os seus índices de adaptabilidade, de $k = 1$ para o menos adaptado à $k = L$ para o mais adaptado se for um problema de minimização, e o contrário para um problema de maximização. Se ocorrer de dois ou mais bits apresentarem o mesmo valor para o índice de adaptabilidade, então ordene-os aleatoriamente com distribuição uniforme.
6. Escolha com igual probabilidade um bit candidato i para sofrer mutação (mudar de 0 para 1 ou de 1 para 0). Gere um número aleatório RAN com distribuição uniforme no intervalo $[0, 1]$. Se $P(k) \approx k^{-\tau}$ for maior ou igual a RAN , o bit é modificado. Caso contrário, o processo é repetido até que um bit seja modificado. Para o bit escolhido para sofrer mutação faça $C = C_i$.
7. Se critério de parada for satisfeito, então retorne a fronteira de Pareto e o conjunto de Pareto, finalizando o algoritmo.
8. Verifique se uma nova execução independente será iniciada. Se sim, retorne ao passo 1. Neste caso um novo ponto no espaço de busca é sorteado. Todavia, a fronteira de Pareto é mantida. Se não, retorne ao passo 2.

Em essência, a estratégia multiobjetivo do M-GEO é a estratégia de um torneio n -para-um, onde, a cada iteração do algoritmo, as $NFOBJ$ funções objetivo competem entre si para serem usadas como função de atribuição de adaptação, porém apenas uma é selecionada.

Como a escolha é totalmente aleatória (com distribuição uniforme), teoricamente, qualquer sequência de escolhas é possível, mesmo aquelas em que apenas uma função objetivo é usada como função de atribuição de adaptação durante toda a busca. Esta aleatoriedade permite ao M-GEO acessar a totalidade da fronteira de Pareto, sendo que a eficiência esperada do algoritmo é imposta pelo parâmetro de ajuste τ sobre todas as funções objetivo (GALSKI, 2006). A Figura 2.14 apresenta uma visão geral do M-GEO sob a forma de fluxograma.

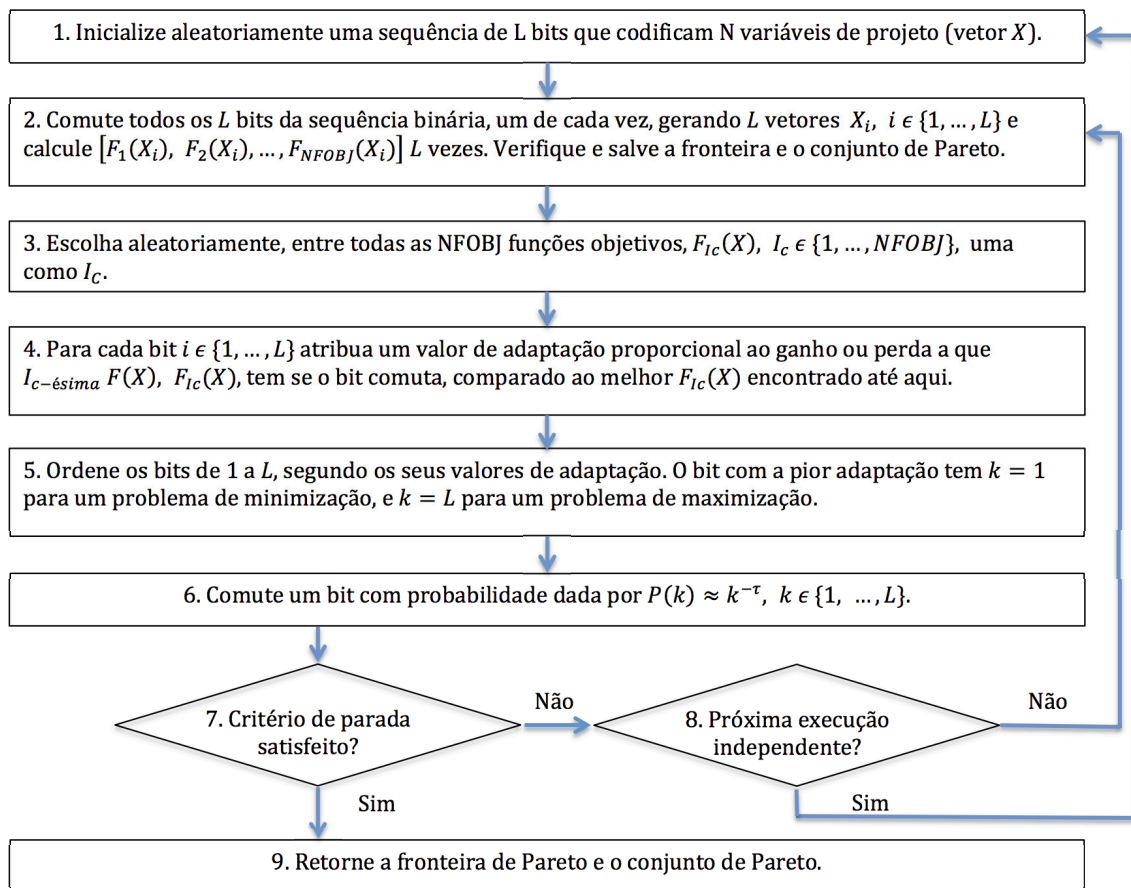


Figura 2.14 - Fluxograma do algoritmo M-GEO.
 Fonte: Adaptado de Galski (2006).

3 O PROBLEMA DA ESCOLHA ÓTIMA DOS LEDs

Neste capítulo é apresentado o processo de modelagem dos LEDs usando os conceitos e equações apresentadas no Capítulo 2. É mostrado a solução do problema de escolha ótima dos LEDs com o algoritmo de Levenberg-Marquardt, que é um algoritmo de otimização local. Em seguida é apresentada a escolha dos LEDs com o algoritmo GEO, e por fim, a escolha ótima dos LEDs conseguida por meio do algoritmo multiobjetivo M-GEO.

3.1 Processo de modelagem dos LEDs

Inicialmente foram selecionados diversos LEDs cobrindo a região espectral de 360 a 1000 nm. Para cada LED selecionado foi obtido o *datasheet* junto ao fabricante para extração das informações importantes e necessárias na modelagem dos LEDs. Não há um padrão nas nomenclaturas utilizadas nos *datasheets*, o que dificultou o processo da identificação e coleta das informações.

Foi coletado de cada LED, quando disponível no *datasheet*, os valores para a intensidade luminosa (mcd), ângulo de visão (graus), comprimento de onda de pico (nm), FWHM (nm), potência radiante (mW) e intensidade radiante (mW/sr). O Apêndice A apresenta a Tabela A.1 com os 160 LEDs candidatos a serem utilizados na FLS.

A modelagem foi realizada calculando inicialmente a DPE normalizada de cada LED, $S_{LED}(\lambda)$, utilizando a Equação 2.11. Em seguida a distribuição de intensidade de radiação por ângulo, $I(\theta_x, \theta_y)$, foi calculada utilizando as Equações 2.12 e 2.13. Para calcular a DPE real de cada LED foi necessário calcular o fator de conversão F_{LED} , que representa o fator multiplicador para converter a DPE normalizada em DPE real.

Alguns LEDs trazem em seu *datasheet* a informação de fluxo radiante Φ_e em mW. Para esses LEDs, o fator de conversão F_{LED} foi calculado baseando-se na Equação 2.16 da seguinte forma:

$$F_{LED} = \frac{\Phi_e}{\int_{\lambda} S_{LED}(\lambda) d\lambda}. \quad (3.1)$$

Para os LEDs que apresentam em seus *datasheets* apenas a intensidade luminosa, I_v , foi calculado o fluxo luminoso utilizando as Equações 2.14 e 2.15 e o fator de

conversão F_{LED} calculado da seguinte forma:

$$F_{LED} = \frac{\Phi_v}{Km \int_{\lambda} S_{LED}(\lambda) V(\lambda) d\lambda}. \quad (3.2)$$

Para os LEDs que apresentam em seus datasheets apenas a intensidade radiante, foi calculado o fluxo radiante utilizando as Equações 2.14 e 2.15 e o fator de conversão F_{LED} foi calculado utilizando a Equação 3.1. Ao final desse processo, tendo calculado o fator de conversão F_{LED} , foi calculado a DPE real para cada um dos LEDs selecionados da seguinte forma:

$$DPE_{LED}(\lambda) = S_{LED}(\lambda) F_{LED}. \quad (3.3)$$

A Figura 3.1 apresenta uma visão geral da modelagem dos LEDs na forma de fluxograma.

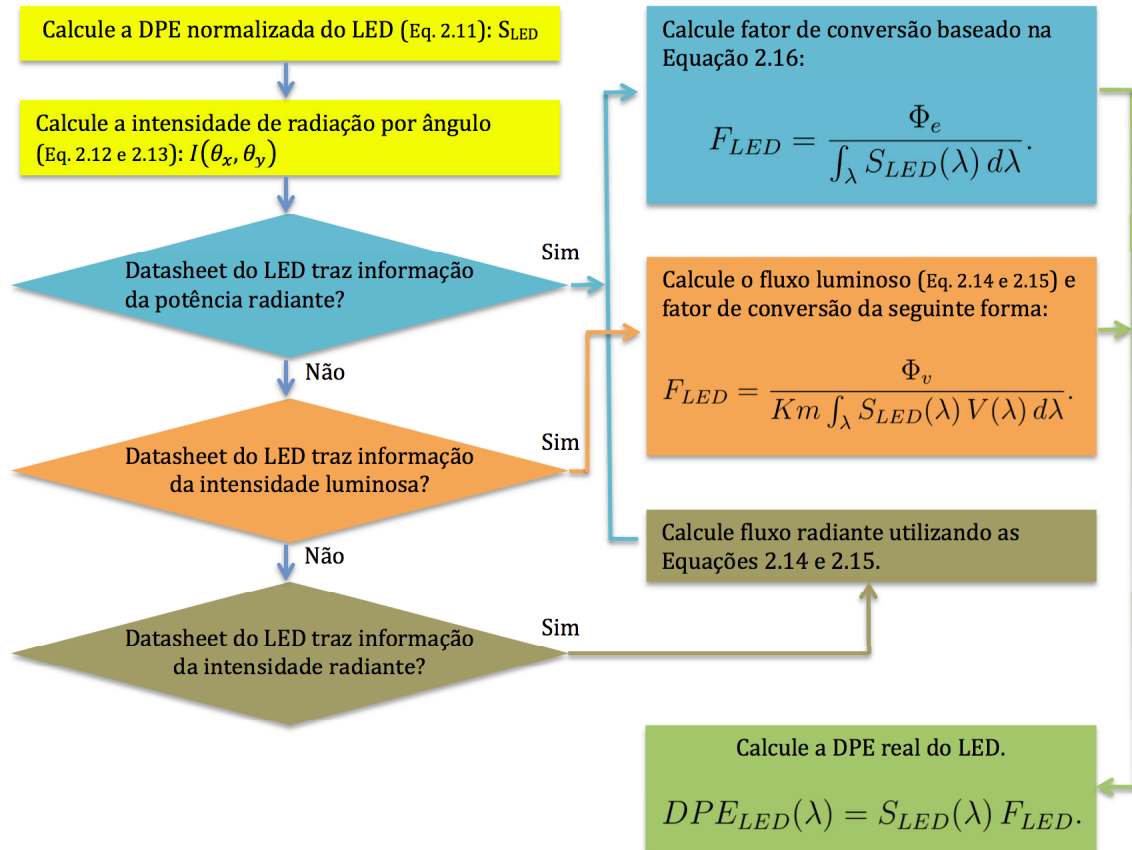


Figura 3.1 - Fluxograma para modelagem dos LEDs.

Para facilitar a tarefa de modelagem dos LEDs, foi implementado um software utilizando como plataforma de desenvolvimento o MATLAB. A Figura 3.2 apresenta a interface gráfica do software. O código fonte do software para modelar os LEDs estará disponível para download em <https://github.com/marcoseborges>, e as principais rotinas implementadas em MATLAB estão disponíveis para consulta no Apêndice B. A Figura 3.3 apresenta a DPE dos 160 LEDs candidatos a serem utilizados na FLS.

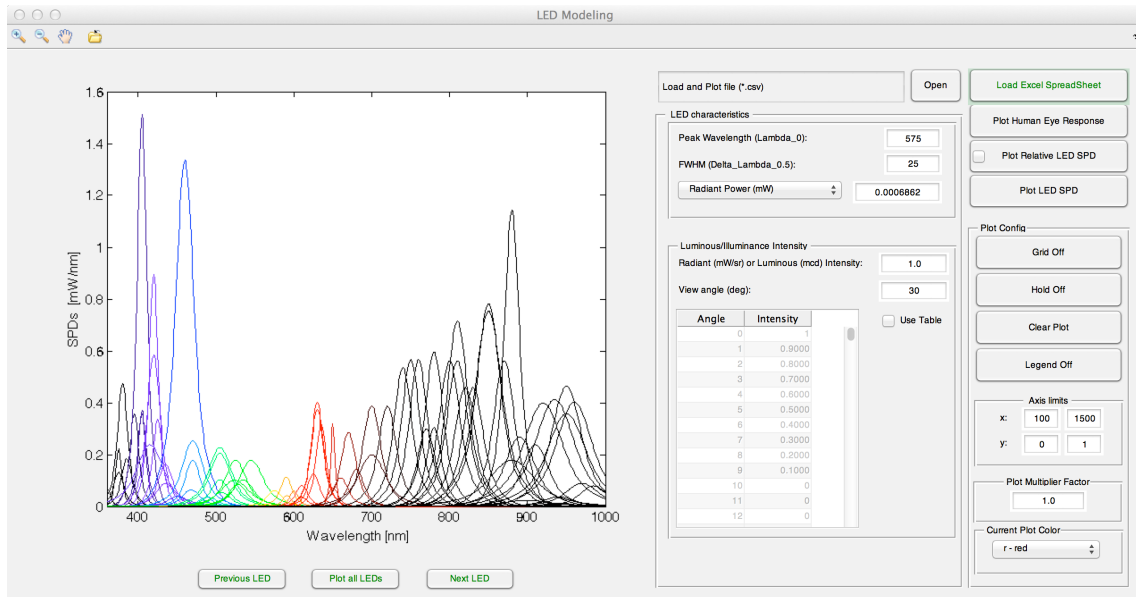


Figura 3.2 - Interface gráfica do Software em MATLAB utilizado para modelar os LEDs.

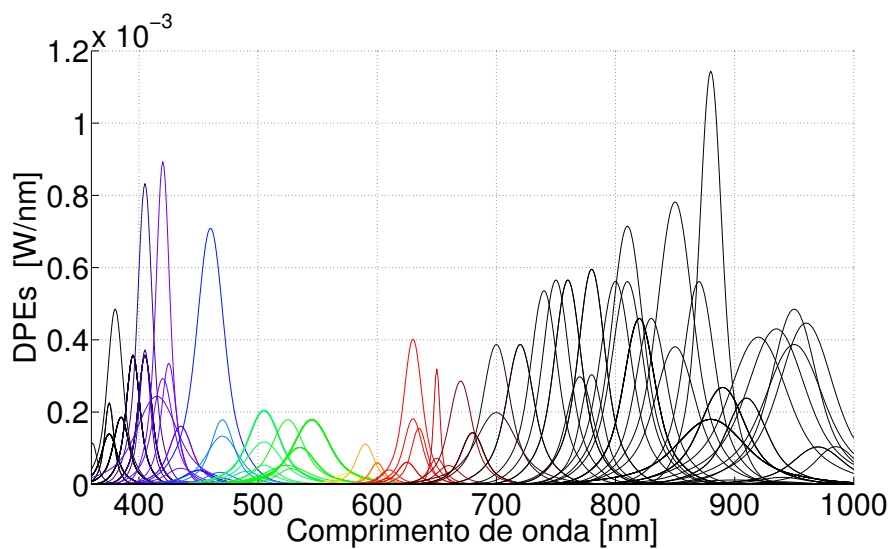


Figura 3.3 - DPE de 160 LEDs candidatos.

3.2 Escolha dos LEDs com Levenberg-Marquardt

A FLS foi projetada para simular as estrelas mais brilhantes, de magnitude zero, até as menos brilhantes, de magnitude cinco. Utilizando as Equações 2.1 a 2.5, foi conseguido o espectro de todas as estrelas que a FLS deve simular. Para realizar a escolha adequada dos LEDs, é necessário garantir que os LEDs tenham potência suficiente para simular todas as estrelas de magnitude zero. Conseguindo potência para simular as estrelas mais brilhantes, de magnitude zero, garante-se também potência para simular as estrelas menos brilhantes. A Figura 3.4 apresenta a irradiância espectral de todas as estrelas de magnitude zero, de 3 000 a 20 000K. A Figura 3.5 apresenta o espectro que engloba todas as estrelas de magnitude zero, de 3 000 a 20 000K.

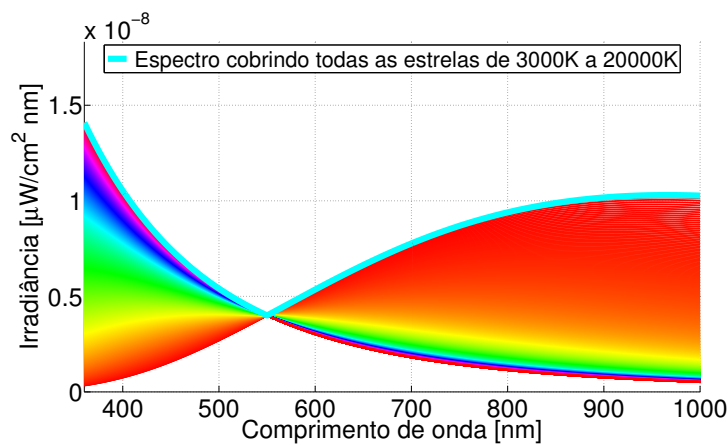


Figura 3.4 - Irradiância espectral de todas as estrelas de magnitude zero, de temperatura de 3 000K a 20 000K selecionadas do catálogo de estrelas Hipparcos.

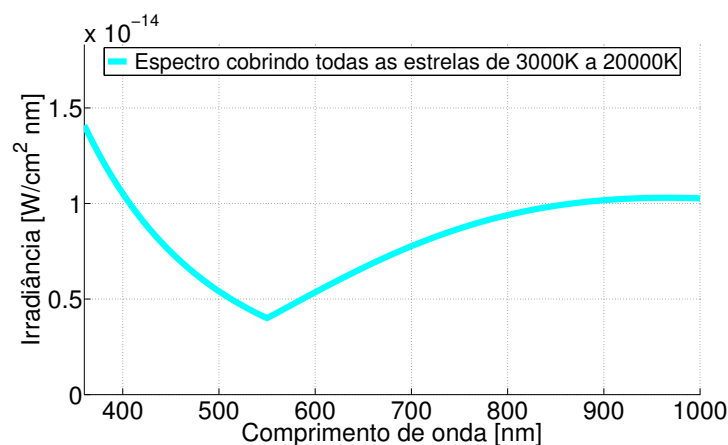


Figura 3.5 - Irradiância espectral cobrindo todas as estrelas de magnitude zero, de temperatura de 3 000K a 20 000K selecionadas do catálogo de estrelas Hipparcos.

Após a modelagem da DPE de cada um dos 160 LEDs, foi desenvolvido um software em MATLAB para simular o comportamento da FLS. Uma função do simulador foi implementada para receber como entrada a potência aplicada em cada um dos 160 LEDs, e devolve como resultado a irradiância espectral gerada na saída do colimador em função dessa potência. A equação abaixo descreve como o cálculo é realizado:

$$I_{FLS}(\lambda, P_{LED}) = \sum_{i=1}^{160} F_{transf} P_{LED_i} DPE_{LED_i}(\lambda), \quad (3.4)$$

onde $I_{FLS}(\lambda, P_{LED})$ é a irradiância espectral esperada na saída do colimador em função da potência aplicada individualmente a cada LED. F_{transf} é a função de transferência do sistema esfera integradora, *pinhole* e colimador, calculada segundo as Equações 2.9 e 2.10. P_{LED_i} é a potência aplicada no LED i (valor real), onde por exemplo, um valor igual a 3,4 significa 3 LEDs em 100% e um quarto LED em 40% da sua potência máxima. Um valor igual a 0,8 representa 1 LED em 80% da sua potência máxima. $DPE_{LED_i}(\lambda)$ é a distribuição de potência espectral do LED i .

Para a escolha ótima dos LEDs, é desejado obter a quantidade e tipos de LEDs com potência suficiente para simular o espectro que engloba todas as estrelas de magnitude zero, de temperatura de 3 000K a 20 000K conforme apresentado na Figura 3.5. A equação abaixo foi utilizada para calcular o erro de ajuste de curva, *fitness*, entre o espectro desejado e o obtido:

$$fitness = \sqrt{\sum_{\lambda=360}^{1000} \left(\sum_{i=1}^{160} F_{transf} P_{LED_i} DPE_{LED_i}(\lambda) - I(\lambda) \right)^2}, \quad (3.5)$$

onde $I(\lambda)$ é a irradiância espectral da estrela desejada na saída do colimador. Para calcular a quantidade total de LEDs apresentada pela solução conseguida após a execução do algoritmo, foi utilizado a seguinte equação:

$$Total_{LEDs} = \sum_{i=1}^{160} [P_{LED_i}]. \quad (3.6)$$

O problema para a escolha ótica dos LEDs é então definido como:

Encontre:

P_{LED_i} , $i \in \{1, \dots, 160\}$, que **minimize** o erro de ajuste de curva, *fitness*, entre o espectro desejado e o obtido.

Sujeito a:

$$0, 0 \leq P_{LED_i} \leq 100, 0$$

O algoritmo de Levenberg-Marquardt (LM) é um método de otimização publicado primeiramente por [Levenberg \(1944\)](#) e aperfeiçoado por [Marquardt \(1963\)](#), sendo considerado um algoritmo muito versátil para estimar parâmetros não lineares ([OZISIK; ORLANDE, 2000](#)). Com o objetivo de realizar a escolha da quantidade e tipos de LEDs necessários para compor a FLS, foi implementado o algoritmo de Levenberg-Marquardt conforme descrito abaixo:

1. Selecione o valor inicial para cada coeficiente de potência $P_{LED_i}^{(0)}$.
2. Calcule o j-ésimo valor de P_{LED} :
 - a) Calcule a diferença entre o espectro desejado e o obtido na saída do colimador para cada comprimento de onda:

$$R(\lambda) = I_{FLS}(\lambda, P_{LED}^{(j-1)}) - I(\lambda). \quad (3.7)$$

- b) Calcule a matriz Jacobiana:

$$J_{I_{FLS}(\lambda)}^{(j-1)}(P_{LED}^{(j-1)}) = \left[\frac{\partial I_{FLS}(\lambda, P_{LED}^{(j-1)})}{\partial P_{LED}^{(j-1)}} \right]. \quad (3.8)$$

- c) j-ésimo valor de P_{LED} :

$$P_{LED}^{(j)} = P_{LED}^{(j-1)} - [(J^T J + \text{diag}(J^T J) \alpha)^{-1} J^T R]. \quad (3.9)$$

3. Se

$$\sum_{\lambda=360}^{1000} \left| I_{FLS}(\lambda, P_{LED}^{(j)}) - I(\lambda) \right| = \sum_{\lambda=360}^{1000} \left| I_{FLS}(\lambda, P_{LED}^{(j-1)}) - I(\lambda) \right| \quad (3.10)$$

então $P_{LED}^{(j)}$ contém o valor final para os coeficientes de potência.

No **passo 1** o valor inicial da potência aplicada em cada um dos tipos de LEDs é definido. Esse pode ser qualquer valor entre 0 a 100. No **passo 2** os próximos valores de potências representados pelo vetor P_{LED} são determinados de forma iterativa. O coeficiente α é o fator de amortecimento do LM, e foi determinado de forma empírica como 0,01. O vetor P_{LED} está fortemente relacionado a potência aplicada aos LEDs. A igualdade é obtida a partir de um grande número de iterações, onde as potências individuais dos LEDs são incrementadas ou decrementadas em pequenos passos. O **passo 3** formula a condição de parada, e pode ser interpretado que o algoritmo termina quando os valores de P_{LED} chegarem a uma solução de estado estacionário.

A Figura 3.6 apresenta o ajuste de curva obtido com LM para o espectro que engloba todas as estrelas de magnitude zero, de temperatura de 3000K a 20000K apresentado na Figura 3.5.

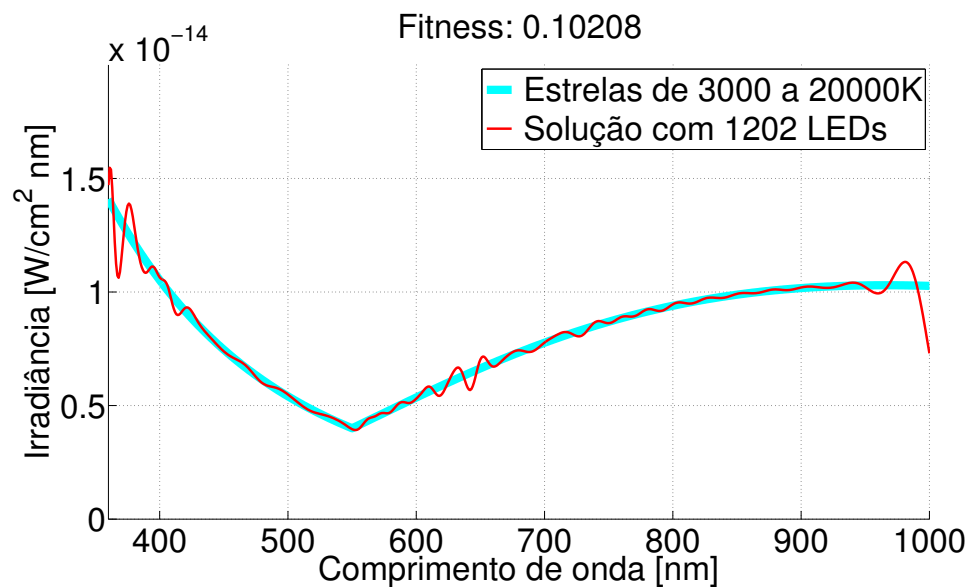


Figura 3.6 - Ajuste de curva para a solução com 1 202 LEDs conseguida com LM.

O problema da solução obtida com LM foi a quantidade enorme de LEDs, o que inviabilizava a construção da FLS, uma vez que não há espaço para mais do que 200 LEDs na esfera integradora, e dessa forma a quantidade de 1 202 LEDs é inviável para a construção da FLS.

Diversos testes foram executados com LM utilizando pontos de inicialização diferentes, porém as soluções obtidas não eram viáveis devido ao elevado número de LEDs. Sendo assim, para conseguir soluções viáveis com LM, foi necessário restringir o número de LEDs, e dessa forma o problema foi reformulado como:

Encontre:

$P_{LED_i}, i \in \{1, \dots, 160\}$, que **minimize** o erro de ajuste de curva, *fitness*, entre o espectro desejado e o obtido.

Sujeito a:

$$0,0 \leq P_{LED_i} \leq 100,0$$

$$Total_{LEDs} \leq 200$$

A Figura 3.7 apresenta o ajuste de curva obtido com LM com a restrição do número máximo de LEDs.

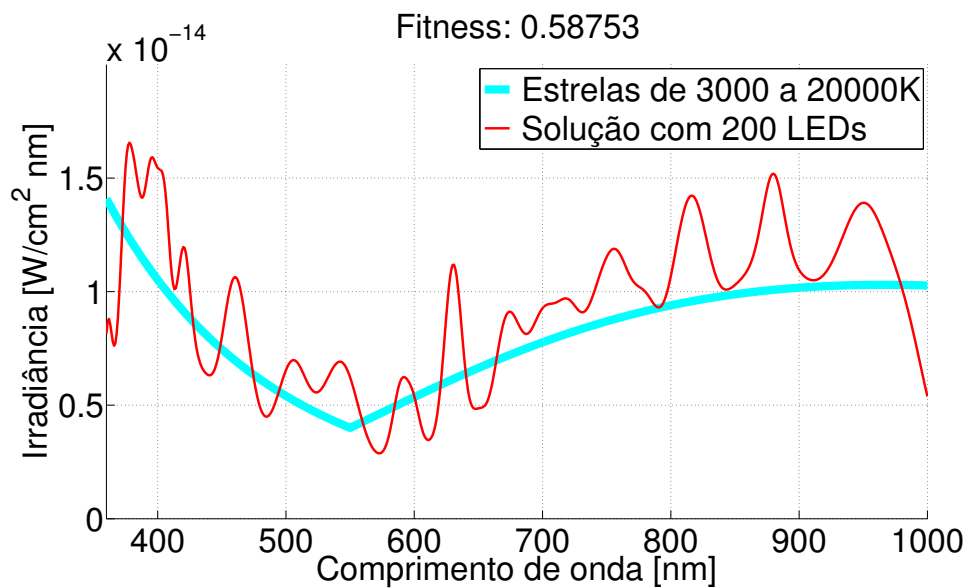


Figura 3.7 - Ajuste de curva para a solução com 200 LEDs conseguida com LM.

A solução conseguida com LM é viável por apresentar a quantidade de LEDs dentro dos limites da FLS, todavia o ajuste de curva não é bom. Por ser um método de otimização local, não há garantias de que a solução conseguida com LM seja uma solução ótima.

3.3 Escolha dos LEDs usando paradigmas de IA

O LM é um método de otimização que procura o mínimo local em uma função e, em geral converge mais rapidamente que um AG. Todavia, este método tende a encontrar soluções sub-ótimas e, em alguns casos, pode não convergir. Para viabilizar o projeto da FLS é necessário minimizar o erro de ajuste de curva e ao mesmo tempo o número de LEDs. Para resolver esse problema multiobjetivo, optou-se por realizar a busca pela solução ótima utilizando paradigmas de IA. Para a tarefa de escolha ótima da quantidade e tipos de LEDs foram implementados os algoritmos GEO e M-GEO.

Dentro do escopo deste trabalho, foram implementados algoritmos em linguagem de programação C++ utilizando a plataforma de desenvolvimento *open source* Qt. Para melhorar o desempenho dos algoritmos e utilizar todo o recurso computacional disponível, o cálculo da função objetivo foi implementado para ser realizado de forma paralela. O código fonte do GEO e do M-GEO será disponibilizado para download em <https://github.com/marcoseborges>.

A codificação do GEO e do M-GEO baseou-se nas descrições apresentadas nos Capítulos 2.5.1 e 2.5.2. Além do cálculo da função objetivo ser realizado em paralelo, outra característica da implementação é a possibilidade de configurar as variáveis de projeto com diferentes restrições, ou seja, é possível definir os valores máximos, mínimos e a precisão de cada variável de forma independente. As restrições das variáveis de projeto são codificadas em um vetor, onde cada posição do vetor corresponde a restrição de uma variável específica. Os algoritmos foram implementados na forma de bibliotecas, e podem ser facilmente usados em outros problemas, necessitando apenas codificar a função objetivo.

Na tentativa de conseguir realizar a escolha ótima dos LEDs usando paradigmas de IA, primeiramente tentou-se resolver o problema com o GEO. Cada LED modelado foi tratado como uma variável de projeto X_i . O valor mínimo para cada variável de projeto foi definido como $X_{min} i = 0$, e valor máximo como $X_{max} i = 30$. O valor da precisão de cada variável foi definido como $e_i = 0,1$.

Outro aspecto importante é a definição do número de bits de cada variável X_i , em função da precisão mínima, e_i , desejada para cada variável. A Equação 2.26 foi utilizada para calcular o menor número de bits, l_i , necessário para representar cada variável, obtendo o número de 9 bits por variável. Neste trabalho todas as variáveis possuem o mesmo valor mínimo e máximo, e portanto foram representadas com a

mesma quantidade de bits. Todavia o algoritmo implementado permite trabalhar com configurações de valores mínimos e máximos diferentes por variável.

Na inicialização do GEO ocorre o preenchimento de uma sequência binária C de comprimento L , por meio de L sorteios de 0s e 1s, ambos com igual probabilidade de ocorrência. Esta sequência binária define uma configuração ou solução candidata a ótimo global, representada no sistema numérico utilizado internamente pelo algoritmo GEO (sistema binário).

Como o sistema numérico utilizado para representar as variáveis de projeto não é o sistema binário e sim decimal, a fim de calcular a função objetivo, $F(X)$, é necessário primeiro converter C em X , ou seja, converter cada variável de projeto, $X_i, i \in \{1, \dots, 160\}$, de sua codificação binária para seu valor decimal correspondente utilizando a Equação 2.27. Assim, obtém-se o valor em decimal de cada variável da solução candidata a ótimo global. O valor decimal de cada variável representa o valor de potência aplicado em cada LED. Por exemplo: um valor igual a 0,80 representa um LED ligado em 80% da sua potência máxima, e um valor igual a 3,40 representa três LEDs ligados em potência máxima e um quarto LED ligado em 40% da sua potência máxima. Um valor igual a 0,85 representa um único LED ligado em 85% da sua potência máxima.

O GEO é um algoritmo de otimização mono-objetivo. Sendo assim, nessa primeira etapa foi definido apenas um único objetivo: a minimização do erro de ajuste de curva (*fitness*). Utilizando o simulador da a FLS, o erro de ajuste de curva entre o espectro obtido e o desejado na saída do colimador foi calculado usando a Equação 3.5. Para fins de comparação com o resultado obtido pelo LM, a quantidade total de LEDs da solução gerada pelo GEO foi calculada utilizando a Equação 3.6. O único parâmetro livre do GEO é o τ , que tem grande influência na convergência do algoritmo (DE SOUSA et al., 2003; GALSKI, 2006). Para medir o desempenho do GEO em função do valor τ , foi analisado os resultados do algoritmo com τ de 0,25 a 9, com incrementos de 0,25. O espectro desejado foi o espectro que engloba todas as estrelas de magnitude zero, de temperatura de 3 000K a 20 000K apresentado na Figura 3.5.

Devido ao alto custo computacional para avaliação da função objetivo, o critério de parada utilizado foi a limitação do número de avaliações da função objetivo (NAF) necessário para achar o mínimo global em 1 440 000, que representa 1 000 gerações do algoritmo para a configuração apresentada. Um segundo critério de parada também foi definido para encerrar a busca quando $|F(X)_{melhor} - F(X)| \leq 1 \times 10^{-15}$, onde

$F(X)$ é o valor da função no ponto ótimo e $F(X)_{melhor}$ o menor valor da função encontrado pelo algoritmo ao longo da busca. O problema para a escolha ótima dos LEDs usando o algoritmo GEO é então definido como:

Encontre:

$X_i, i \in \{1, \dots, 160\}$, que **minimize** o erro de ajuste de curva, *fitness*, entre o espectro desejado e o obtido.

Sujeito a:

$$0,0 \leq X_i \leq 30,0$$

Devido à característica estocástica do GEO, para a correta análise da influência do parâmetro τ , foram realizadas 50 execuções independentes do algoritmo para cada valor τ . Na Figura 3.8 é apresentada a influência do τ no desempenho do GEO.

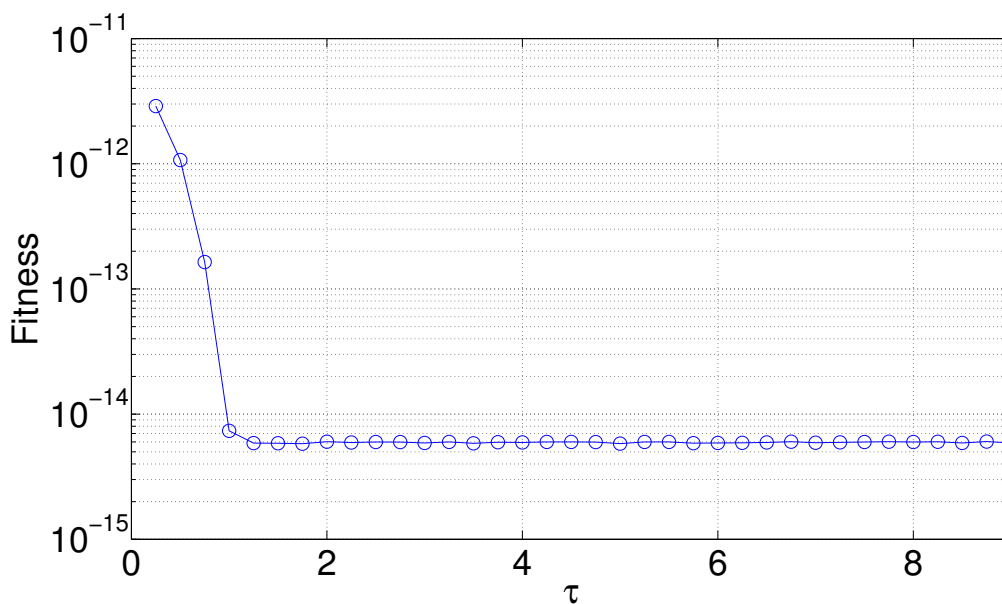


Figura 3.8 - Erro de ajuste de curva médio, *fitness*, em 50 execuções independentes do algoritmo GEO, para identificar o mínimo global em função do τ . Critérios de parada: $|F(X)_{melhor} - F(X)| \leq 1 \times 10^{-15}$ e $NAF > 1\,440\,000$.

Analisando a Figura 3.8 parece não existir um único valor ótimo para τ . No intervalo 0,25 a 0,75 o GEO apresentou seu pior desempenho, porém a partir de $\tau = 1,00$ não apresentou melhorias no ajuste de curva. Analisando os logs gerados pelo algoritmo, foi constatado que em todas as 50 execuções independentes do GEO para todos os

valores de τ , o critério de parada predominante foi o número máximo de avaliação da função objetivo, o que fez necessário uma segunda análise de desempenho.

Essa primeira análise de desempenho do GEO em função do parâmetro τ não apresentou informações satisfatórias para a escolha do melhor valor para τ . Sendo assim, uma segunda análise foi realizada alterando apenas um dos critérios de parada, onde foi definido para encerrar a busca quando $|F(X)_{\text{melhor}} - F(X)| \leq 6 \times 10^{-15}$. Com essa alteração, o GEO passou a encerrar a busca pelo ótimo global muito antes de atingir o número máximo de avaliações da função objetivo.

Essa segunda análise visou encontrar o melhor valor para τ em função da quantidade de avaliações da função objetivo. Na Figura 3.9 é apresentada a influência do τ no desempenho do GEO. Uma característica importante para um algoritmo de otimização é que a convergência para o mínimo se dê de forma rápida. Analisando a Figura 3.9 é perceptível a existência de um valor para τ que fornece os melhores resultados com o menor número de avaliações da função objetivo, e esse valor é 3,75.

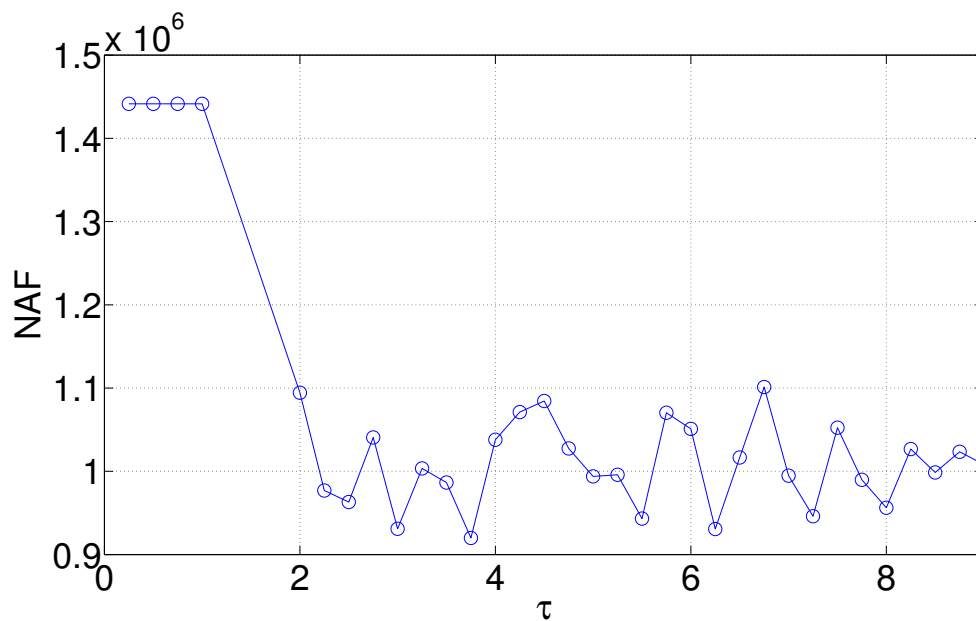


Figura 3.9 - Número médio de avaliações da função objetivo, NAF , em 50 execuções independentes do algoritmo GEO, para identificar o mínimo global em função do τ . Critérios de parada: $|F(X)_{\text{melhor}} - F(X)| \leq 6 \times 10^{-15}$ e $NAF > 1\,440\,000$.

3.3.1 Escolha dos LEDs com GEO

Após o levantamento das informações técnicas e a modelagem dos LEDs, a definição da função de transferência do sistema esfera integradora, *pinhole* e colimador, a modelagem da estrela a ser simulada, e a definição da função objetivo, foi executado o algoritmo GEO, implementado em C++, rodando em uma máquina Windows 64 bits, processador Intel i7-3770 3,40GHz com memória RAM de 8 GB. O parâmetro de ajuste livre do GEO, τ , foi definido como 3,75 conforme análise de desempenho do algoritmo. A Figura 3.10 apresenta o ajuste de curva obtido com o GEO para a interpolação de estrelas de 3000 a 20000K.

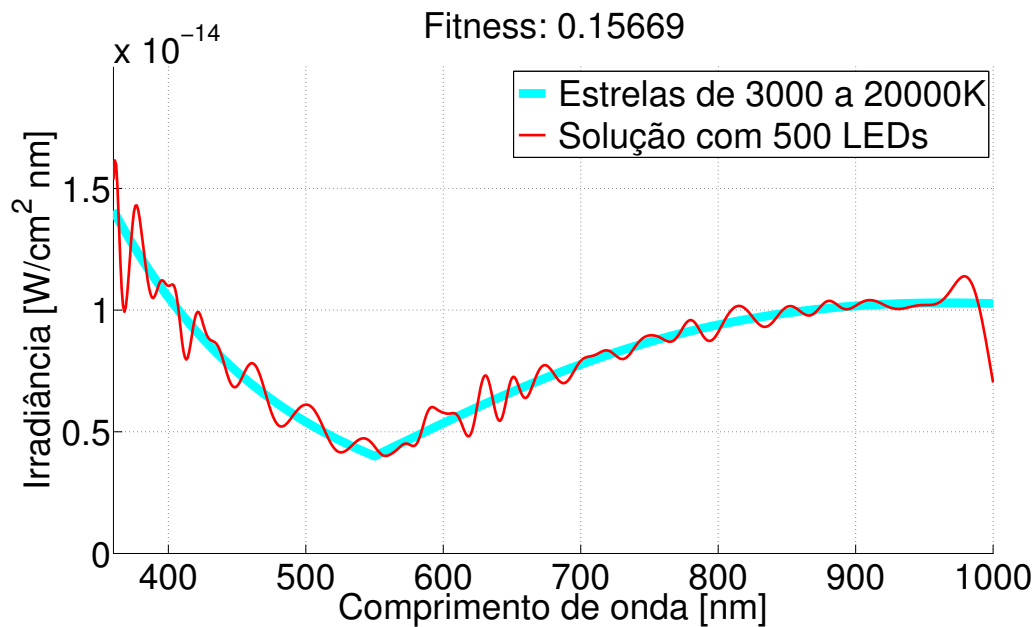


Figura 3.10 - Ajuste de curva para a solução com 500 LEDs conseguida com GEO.

A quantidade de 500 LEDs ainda é inviável para a construção da FLS, todavia menos da metade da quantidade apresentada pelo LM sem restrições. O erro de ajuste de curva do GEO é próximo ao apresentado pelo LM com 1202 LEDs.

3.3.2 Escolha dos LEDs com M-GEO

Usando o GEO conseguiu-se resultados com ajuste de curva bem próximo ao apresentado pelo LM. Entretanto, a quantidade de LEDs ainda ultrapassava os limites impostos pela estrutura mecânica da FLS, que comporta no máximo 200 LEDs.

Sendo assim, foi implementado a versão multiobjetivo do GEO, o M-GEO. A configuração do M-GEO foi bem parecida com a utilizada no GEO: o valor mínimo para cada variável de projeto X_i , foi definido como $X_{min} i = 0,0$, e valor máximo como $X_{max} i = 100,0$. O valor da precisão de cada variável foi definido como $e_i = 0,1$. Assim como no GEO, o parâmetro de ajuste livre do M-GEO, τ , também foi definido como 3,75. Por último foram definidos dois objetivos: i) a **minimização** do erro de ajuste de curva e ii) a **minimização** da quantidade de LEDs. Matematicamente, o problema para a escolha ótima dos LEDs usando o algoritmo M-GEO é então definido como:

Encontre:

$X_i, i \in \{1, \dots, 160\}$, que **minimize**

$$F(X) = \begin{bmatrix} F_1(X) \\ F_2(X) \end{bmatrix} = \begin{bmatrix} \sqrt{\sum_{\lambda=360}^{1000} \left(\sum_{i=1}^{160} F_{transf} X_i DPE_{LED_i}(\lambda) - I(\lambda) \right)^2} \\ \sum_{i=1}^{160} [X_i] \end{bmatrix}. \quad (3.11)$$

Sujeito a:

$$0,0 \leq X_i \leq 100,0$$

Onde:

F_{transf} = função de transferência do sistema esfera integradora, *pinhole* e colimador, calculada segundo as Equações 2.9 e 2.10.

DPE_{LED_i} = distribuição de potência espectral do LED i .

$I(\lambda)$ = irradiância espectral da estrela desejada na saída do colimador.

Na Equação 3.11, $F_1(X)$ é o erro de ajuste de curva entre o espectro desejado e o obtido: *fitness*. Já $F_2(X)$ representa a quantidade total de LEDs: $Total_{LEDs}$. Dessa forma, $F(X)$ é uma função vetorial com dois elementos, $F(X) = [F_1(X), F_2(X)]^T$, e o objetivo é minimizar $F_1(X)$ e $F_2(X)$, obtendo a fronteira de Pareto e as respectivas soluções para a escolha dos LEDs. As 160 variáveis de projeto, ou seja, os 160 tipos de LEDs são representados pelo vetor X no processo de otimização.

Assim como o GEO, o algoritmo M-GEO também foi implementado em C++, e executado em uma máquina Windows 64 bits, processador Intel i7-3770 3,40GHz com memória RAM de 8 GB. A Figura 3.11 apresenta a Fronteira de Pareto obtida através da execução do algoritmo, consumindo aproximadamente 77 horas.

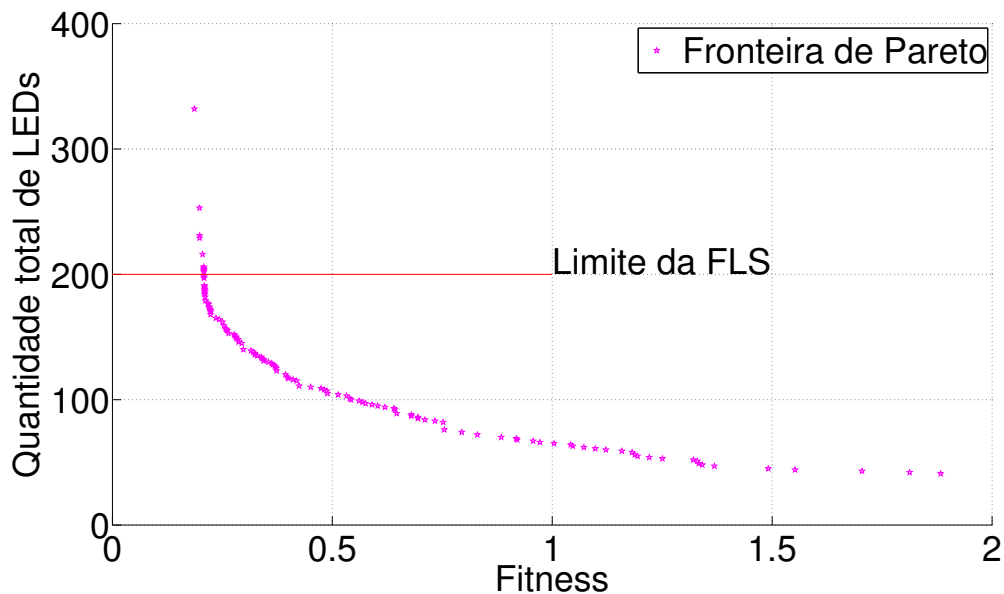


Figura 3.11 - Fronteira de Pareto gerada pelo M-GEO.

Como esperado, o algoritmo gerou diferentes soluções viáveis, permitindo escolher a melhor solução de acordo com as limitações do projeto. Analisando a Fronteira de Pareto e considerando as soluções abaixo do limite de LEDs da FLS, há várias soluções viáveis com a quantidade de 175 a 200 LEDs com *fitness* muito próximo. As Figuras 3.12 e 3.13 apresentam soluções com a quantidade de LEDs acima do limite da FLS.

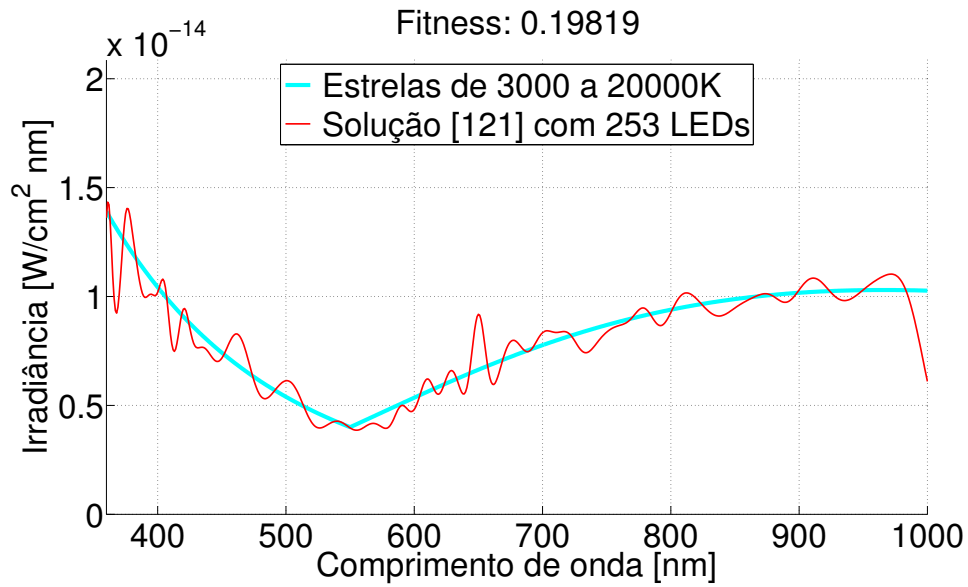


Figura 3.12 - Ajuste de curva para a solução com 253 LEDs conseguida com M-GEO.

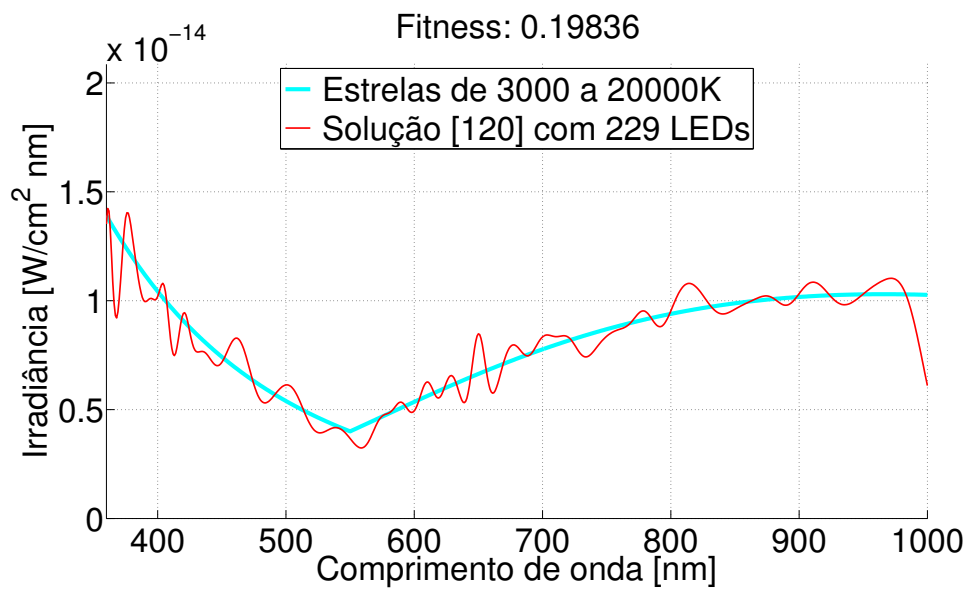


Figura 3.13 - Ajuste de curva para a solução com 229 LEDs conseguida com M-GEO.

Quanto menor o valor do *fitness*, melhor é a solução. As Figuras 3.14 e 3.15 apresentam soluções com a quantidade viável de LEDs. O ajuste de curva gerado pelas soluções com 253 e 229 LEDs são bem próximos aos gerados pelas soluções com 198 e 174 LEDs.

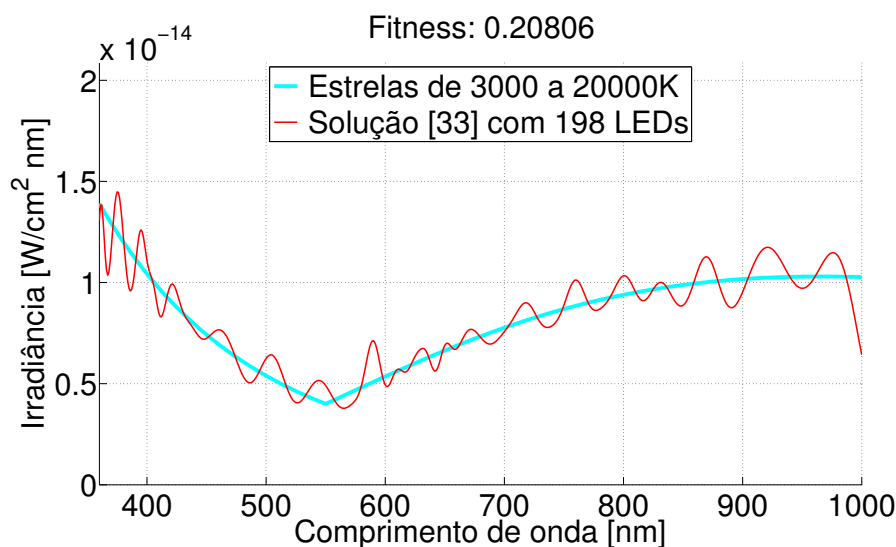


Figura 3.14 - Ajuste de curva para a solução com 198 LEDs conseguida com M-GEO.

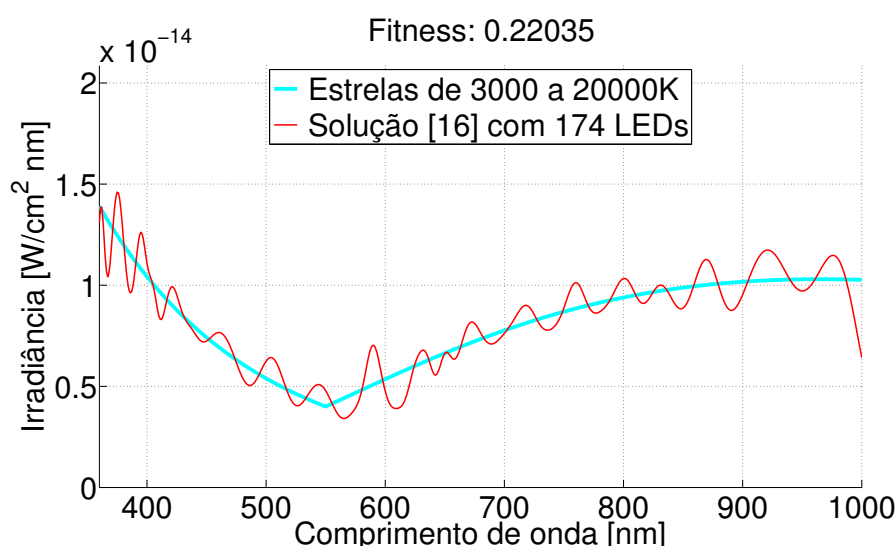


Figura 3.15 - Ajuste de curva para a solução com 174 LEDs conseguida com M-GEO.

Foi escolhida para a construção da FLS a solução com 198 LEDs. O número máximo de LEDs que a FLS comporta é 200 LEDs. Outras soluções também poderiam ser escolhidas, todavia, todas as soluções com menos de 198 LEDs apresentam *fitness* pior. Antes da compra dos LEDs, foram realizadas diversas simulações para verificar a solução conseguida por meio do M-GEO. A Figura 3.16 apresenta o resultado da simulação para uma estrela de magnitude zero a 7500K. Embora o ajuste de curva não seja perfeito, a pequena diferença entre o espectro desejado e a ser fornecido pela FLS não é significativa para calibração do SEA.

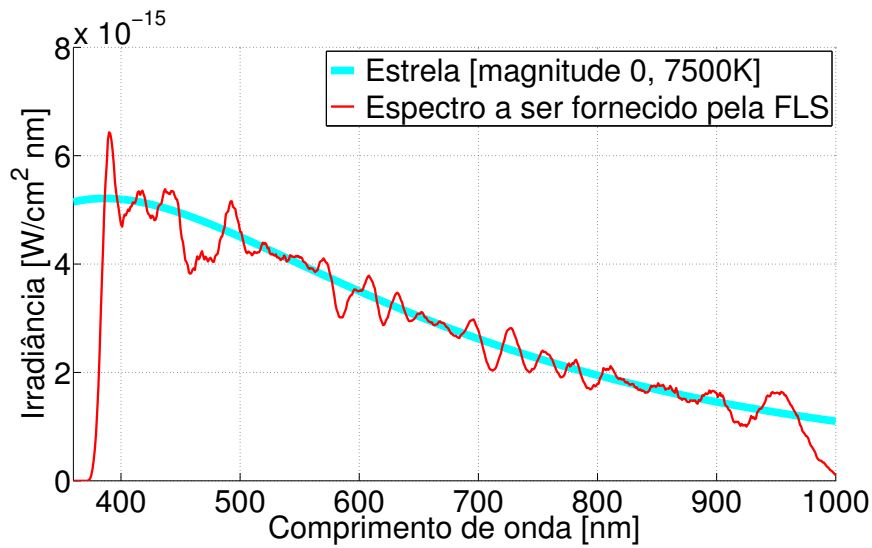


Figura 3.16 - Simulação da FLS para uma estrela de magnitude zero a 7500K.

A Figura 3.17 apresenta o resultado da simulação para uma estrela de magnitude zero a 20000K. Na simulação dessa estrela ocorreu um comportamento similar ao da Figura 3.16, com um ajuste de curva pobre em 480 a 490 nm. Diversas soluções foram avaliadas para melhorar o ajuste de curva nessa região, porém ao adicionar LEDs para melhorar o resultado nessa região, era necessário reduzir a quantidade de LEDs de outras regiões, o que não trazia melhorias.

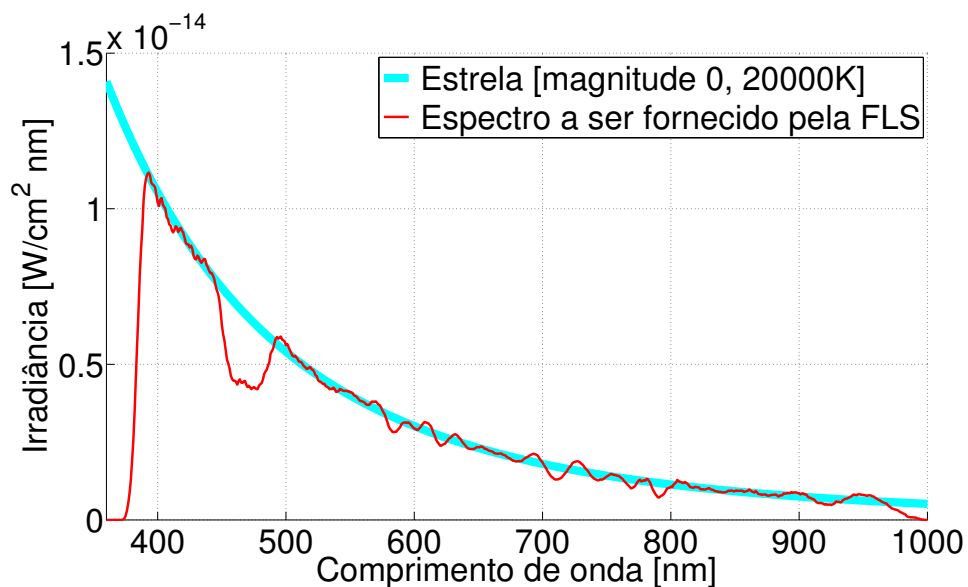


Figura 3.17 - Simulação da FLS para uma estrela de magnitude zero a 20000K.

A Figura 3.18 apresenta o resultado da simulação para uma estrela de magnitude zero a 5600K. Os ajustes de curvas não são perfeitos devido ao limite da quantidade de LEDs imposto pela FLS. Mesmo assim atendem as exigências do SEA.

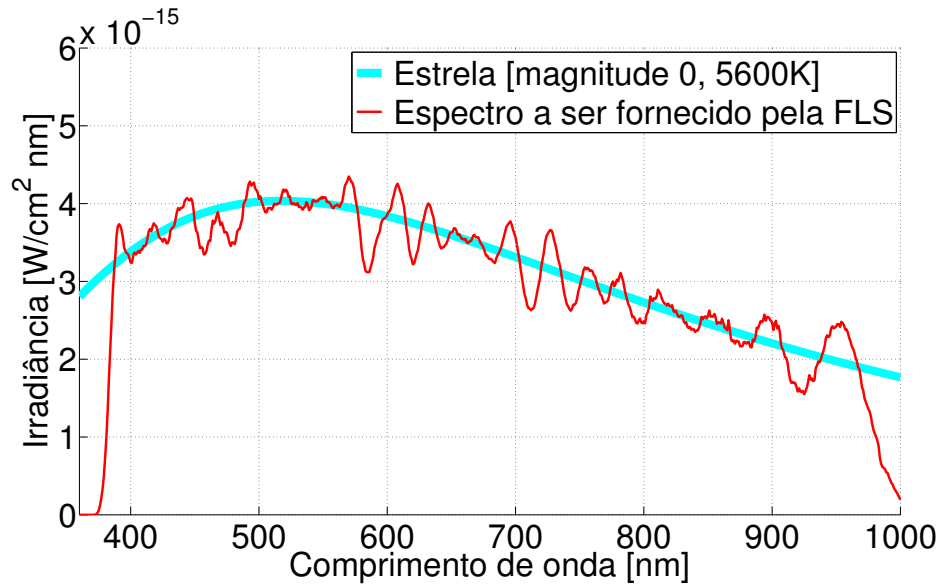


Figura 3.18 - Simulação da FLS para uma estrela de magnitude zero a 5600K.

4 O SOFTWARE DE CONTROLE DA FLS

Após a realização da escolha adequada e a compra dos LEDs para compor a FLS, os mesmos foram montados nas cabeças de fixação de LEDs que são acopladas na esfera integradora. Também, foi realizado o levantamento das características espectrais dos LEDs utilizando um espectroradiômetro para coleta das informações. Com os dados espectrais dos LEDs coletados, foi implementado em linguagem de programação C++ um sistema de controle com retroalimentação capaz de controlar a FLS e assim gerar espectros das estrelas para a calibração do SEA.

Os software foi desenvolvido em C++ utilizando a plataforma de desenvolvimento Qt 5, e é composto basicamente por três módulos: i) o módulo responsável pela comunicação com a FLS e controle da potência individual dos LEDs, ii) o módulo responsável pelo controle do espectroradiômetro para leitura do espectro gerado dentro esfera integradora e iii) o módulo de controle e simulação de estrelas.

O sistema desenvolvido para simular estrelas controla a potência individual dos LEDs da FLS gerando assim um espectro resultante da contribuição espectral de todos os LEDs acoplados na esfera integradora. Esse espectro resultante é então obtido por meio do espectroradiômetro, que é o *feedback* do sistema. O módulo que é responsável pela simulação em si, compara o espectro obtido com o espectro desejado e assim corrige a potência aplicada em cada LED de forma a minimizar o erro de ajuste de curva. Esse processo é repetido de forma iterativa até alcançar uma condição de parada.

A FLS permite controlar a potência individual de 96 canais. Em cada canal foram ligados vários LEDs do mesmo tipo em série, de forma a incluir os 198 LEDs escolhidos. O controle da potência de cada canal, ou seja, de cada grupo de LEDs do mesmo tipo, é possível pela comunicação USB com a FLS. A informação da potência desejada em cada canal é enviada de forma serial, onde é possível definir o valor da potência em níveis digitais no intervalo de 0 a 4095 (valores inteiros). O valor 0 representa que o LED está totalmente apagado e 4095 que ele está aceso na potência máxima.

A Figura 4.1 apresenta a interface gráfica do software da FLS com o ajuste de curva conseguido para simular uma estrela de magnitude zero a 7500K. Mais detalhes do software são apresentados no Manual do Software da FLS contido no [Apêndice C](#). O [Apêndice D](#) contém informações referentes a configuração do ambiente de desenvolvimento.

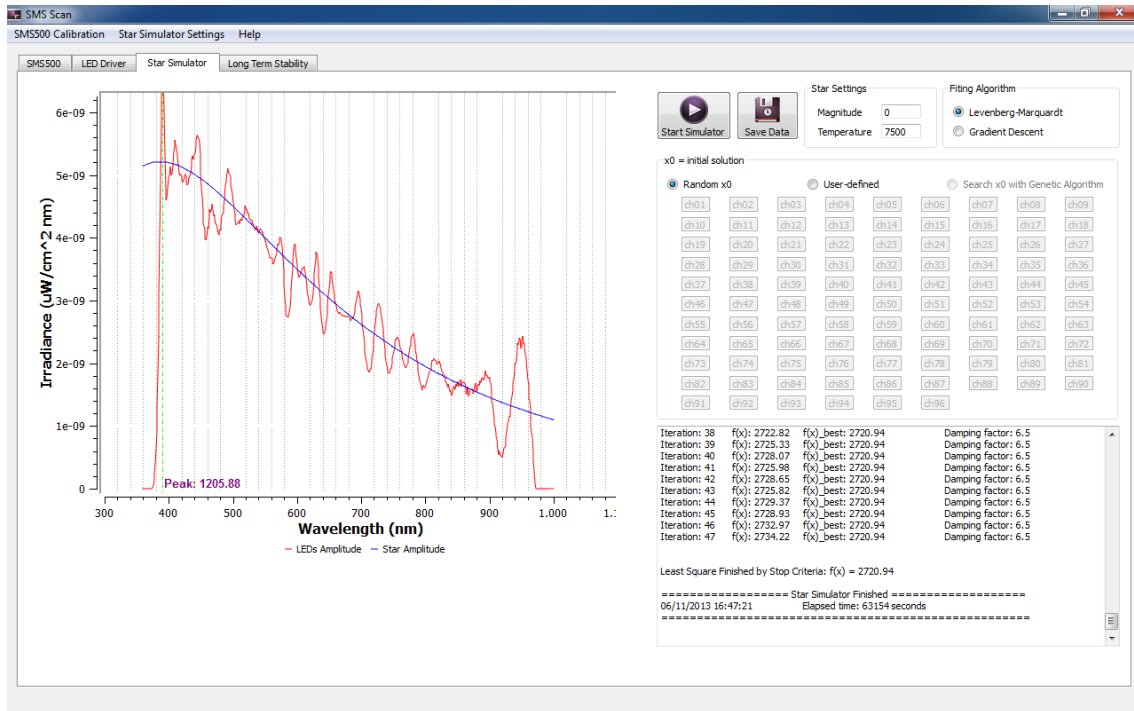


Figura 4.1 - Interface gráfica do Software da FLS, implementado para simular estrelas.

4.1 O algoritmo de Levenberg-Marquardt (LM)

O LM implementado no Capítulo 3.2 foi utilizado na tentativa de realizar a escolha ótima da quantidade e tipos de LEDs, porém seus resultados foram superados pelo M-GEO. Todavia, o LM se demonstrou muito eficiente para ajustar os fatores de potência dos LEDs de forma a gerar um bom ajuste de curva. Sendo assim, o primeiro algoritmo implementado para tornar possível a FLS simular espectros de estrelas foi o LM, conforme descrito no Capítulo 3.2 com algumas pequenas modificações:

1. Selecione o valor inicial para cada coeficiente de potência de cada canal da FLS, $P_{CANAL_i}^{(0)}$, $i \in \{1, \dots, 96\}$.
2. Calcule o j-ésimo valor para o vetor P_{CANAL} :
 - a) Calcule a diferença entre a irradiância espectral gerada pela FLS em função da potência aplicada aos canais e a irradiância espectral da estrela desejada:

$$R(\lambda) = I_{FLS}(\lambda, P_{CANAL}^{(j-1)}) - I(\lambda). \quad (4.1)$$

b) Calcule a matriz Jacobiana:

$$J_{I_{FLS}(\lambda)}^{(j-1)}(P_{CANAL}^{(j-1)}) = \left[\frac{\partial I_{FLS}(\lambda, P_{CANAL}^{(j-1)})}{\partial P_{CANAL}^{(j-1)}} \right]. \quad (4.2)$$

c) j-ésimo valor para o vetor P_{CANAL} :

$$P_{CANAL}^{(j)} = P_{CANAL}^{(j-1)} - [(J^T J + \text{diag}(J^T J) \alpha)^{-1} J^T R]. \quad (4.3)$$

3. Se

$$\sum_{\lambda=360}^{1000} \left| I_{FLS}(\lambda, P_{CANAL}^{(j)}) - I(\lambda) \right| = \sum_{\lambda=360}^{1000} \left| I_{FLS}(\lambda, P_{CANAL}^{(j-1)}) - I(\lambda) \right| \quad (4.4)$$

então $P_{CANAL}^{(j)}$ contém o valor final para os coeficientes de potência.

Uma característica do algoritmo LM implementado, é que foi necessário adquirir a DPE de cada canal da FLS em função da potência aplicada ao canal. A potência de cada canal é configurada pela definição de um valor entre 0 a 4095. Dessa forma, foi coletado para cada um dos 96 canais, a DPE do canal em função do nível digital, de 0 a 4095. Com esses dados experimentais coletados foi criado um banco de dados que é utilizado pelo LM na hora de calcular a matriz Jacobiana. Abaixo é descrito em detalhes cada passo do LM:

No **passo 1** é definido o valor inicial da potência aplicada nos canais. O valor deve ser um número inteiro compreendido entre 0 a 4095.

No **passo 2** os próximos valores para a potência dos canais são determinados de forma iterativa buscando minimizar o erro quadrático do ajuste de curva. Em (a) é calculado para cada comprimento de onda, a diferença entre o espectro obtido pelo espectrorradiômetro e o espectro da estrela desejada. Em (b) é montado a matriz Jacobiana usando a base de dados com as medições dos LEDs, ou seja, as derivadas parciais são calculadas usando os valores das DPEs dos canais previamente medidos. O comprimento de onda λ que a FLS opera é de 360 a 1 000 nm, e o número de canais é 96, dessa forma a Jacobiana é uma matriz 641×96 . Em (c) é calculado os novos valores para a potência de cada canal. O coeficiente α é o fator de amortecimento do LM, e no software da FLS foi definido de forma empírica como 6,5.

O **passo 3** formula a condição de parada, que encerra a iteração do algoritmo quando os valores de P_{CANAL} chegam a uma solução de estado estacionário.

Diversos testes com diferentes estrelas foram realizados com o LM, que apresentou resultados satisfatórios. A Figura 4.2 apresenta o ajuste de curva para uma estrela de magnitude zero a 3500K conseguido com o algoritmo LM implementado no software da FLS. O ajuste de curva não é perfeito mas atende as exigências para calibração do SEA. O resultado foi conseguido com 132 iterações do algoritmo iniciado com valores aleatórios. O tempo de convergência foi de aproximadamente 3 minutos.

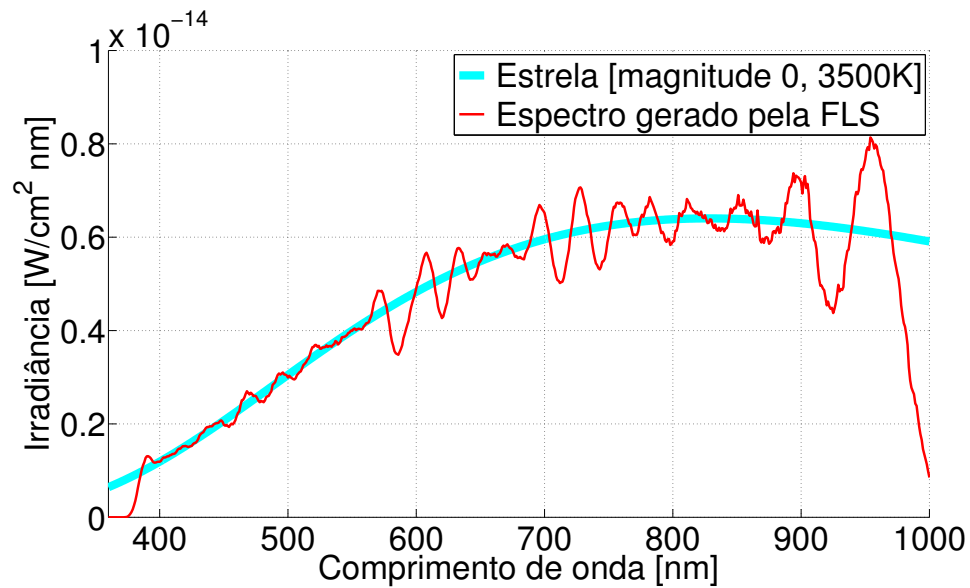


Figura 4.2 - Simulação de estrela de magnitude zero a 3500K conseguido com LM.

A Figura 4.3 apresenta o ajuste de curva para uma estrela de magnitude zero a 5800K. O resultado foi conseguido com 191 iterações do algoritmo consumindo aproximadamente 4 minutos para convergência.

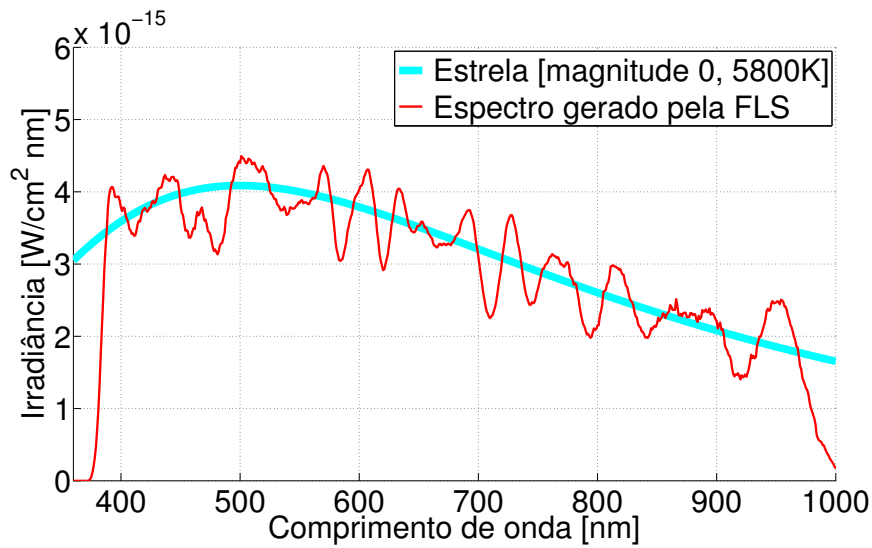


Figura 4.3 - Simulação de estrela de magnitude zero a 5 800K conseguido com LM.

A Figura 4.4 apresenta o ajuste de curva para uma estrela de magnitude zero a 12 130K. O resultado foi conseguido com 69 iterações consumindo aproximadamente 1,5 minutos para convergência. O ajuste de curva não é muito bom perto de 480 a 490nm já era previsto desde as simulações antes da compra dos LEDs. Esse problema é causado devido as limitações da quantidade de LEDs imposta pela FLS.

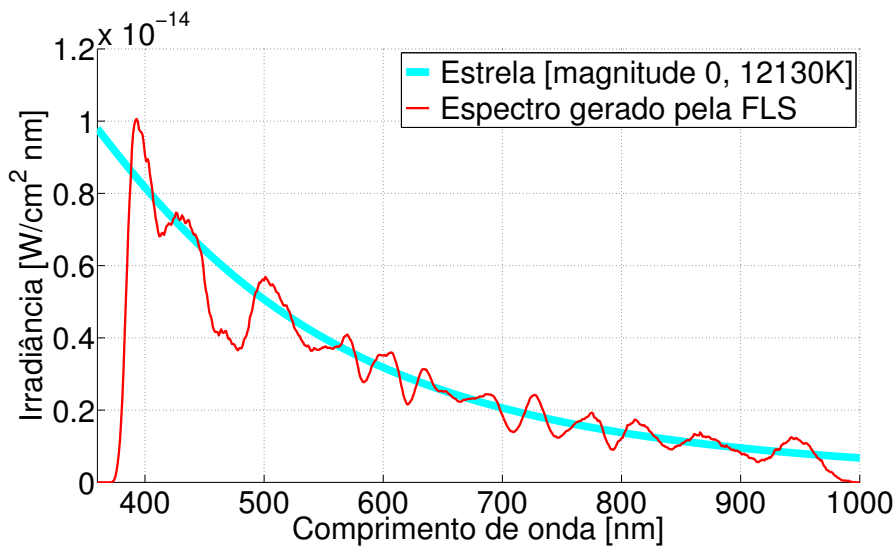


Figura 4.4 - Simulação de estrela de magnitude zero a 12 130K conseguido com LM.

4.2 O algoritmo do Gradiente Descendente (GD)

O segundo algoritmo implementado para a FLS simular espectros de estrelas foi o GD, conforme descrito por Fryc et al. (2004), Fryc et al. (2005a) e Fryc et al. (2005b):

1. Selecione o valor inicial para cada coeficiente de potência de cada canal da FLS, $P_{CANAL_i}^{(0)}$, $i \in \{1, \dots, 96\}$.
2. Calcule o j -ésimo valor de P_{CANAL_i} :

$$P_{CANAL_i}^{(j)} = P_{CANAL_i}^{(j-1)} - \alpha \frac{\partial \sum_{\lambda=360}^{1000} \left(\sum_{i=1}^{96} I_{FLS}(\lambda, P_{CANAL_i}^{(j-1)}) - I(\lambda) \right)^2}{\partial P_{CANAL_i}^{(j-1)}} \quad (4.5)$$

3. Se

$$\sum_{\lambda=360}^{1000} \left| I_{FLS}(\lambda, P_{CANAL}^{(j)}) - I(\lambda) \right| = \sum_{\lambda=360}^{1000} \left| I_{FLS}(\lambda, P_{CANAL}^{(j-1)}) - I(\lambda) \right|, \quad (4.6)$$

então $P_{CANAL}^{(j)}$ é o valor final do coeficiente.

Uma diferença do GD em relação ao LM, é que o GD não precisa do banco de dados com as medições das DPEs dos canais da FLS em função da potência aplicada aos canais. A cada iteração do LM, os valores de potência de todos os LEDs são atualizados simultaneamente, e uma única medição com o espectroradiômetro é realizada para obter espectro gerado pela nova configuração de potência.

Diferentemente do LM, o GD atualiza a potência dos canais um por vez, e para cada canal com a potência atualizada, uma medição com o espectroradiômetro é realizada, o que o torna mais lento em relação ao LM. Ao implementar o GD esperava-se conseguir resultados melhores que o LM, porém os resultados para o ajuste de curva foram bem próximos aos conseguidos com LM. Embora o GD tenha apresentado resultados semelhantes aos conseguidos com o LM, ele apresentou uma grande desvantagem em relação ao tempo para convergência. Enquanto o LM leva minutos para convergir, o GD para conseguir resultado semelhante consome várias horas.

Para fins de comparação entre os algoritmos LM e GD, foram realizados testes de ajuste de curva para as mesmas simulações com o LM. A Figura 4.5 apresenta o ajuste de curva para uma estrela de magnitude zero a 3500K conseguido com o

algoritmo GD implementado no software da FLS. O resultado foi conseguido com 182 iterações do algoritmo iniciado com valores aleatórios. O tempo de convergência foi de aproximadamente 7,5 horas.

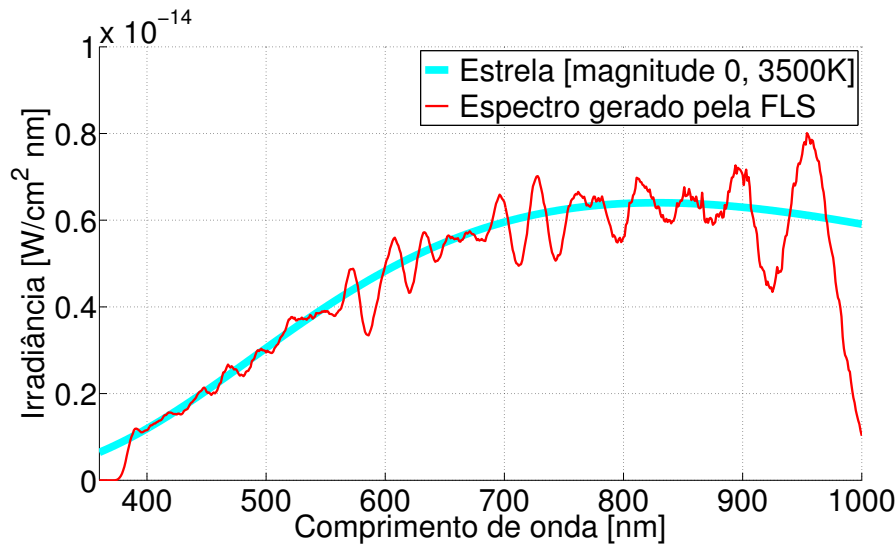


Figura 4.5 - Simulação de estrela de magnitude zero a 3 500K conseguido com GD.

O tempo elevado para convergência do GD é devido ao tempo que o espectroradiômetro leva para obter a resposta espectral a cada correção no valor da potência do canal da FLS. Em outras palavras, a cada iteração do algoritmo são calculadas 96 derivadas parciais (uma derivada parcial por canal), e para cada derivada calculada é realizado a leitura do espectro gerado na esfera integradora. Em geral, o espectroradiômetro consome alguns milissegundos para realizar a leitura do espectro, todavia esse processo em grande quantidade consome muito tempo.

O LM também calcula várias derivadas, todavia é mais rápido porque utiliza da base de dados que contém as DPEs dos canais da FLS em função da potência aplicada ao canal. A Figura 4.6 apresenta o ajuste de curva para uma estrela de magnitude zero a 5 800K. O resultado foi conseguido com 250 iterações do algoritmo consumindo aproximadamente 10 horas para convergência.

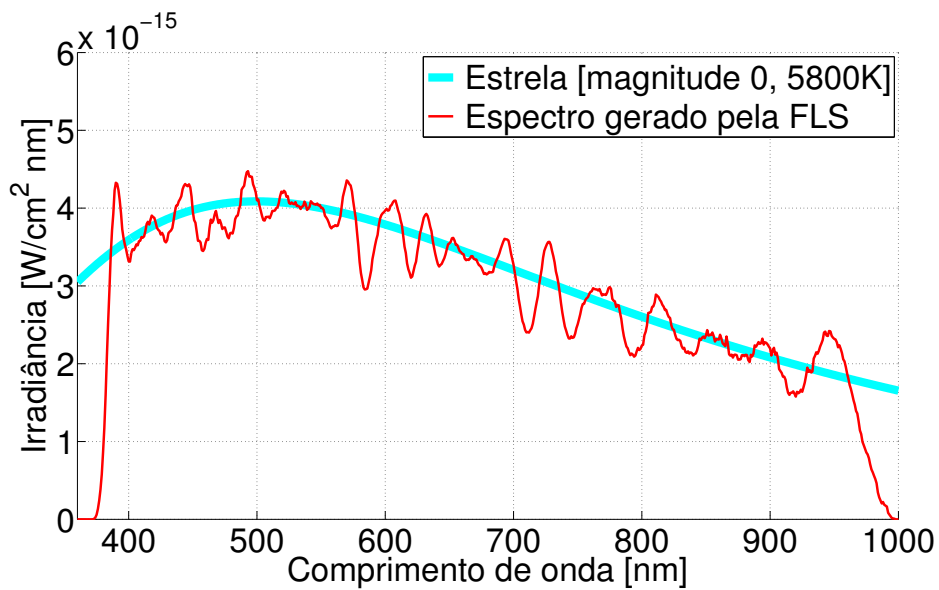


Figura 4.6 - Simulação de estrela de magnitude zero a 5 800K conseguido com GD.

A Figura 4.7 apresenta o ajuste de curva para uma estrela de magnitude zero a 12 130K. O resultado foi conseguido com 120 iterações consumindo aproximadamente 5 horas para convergência.

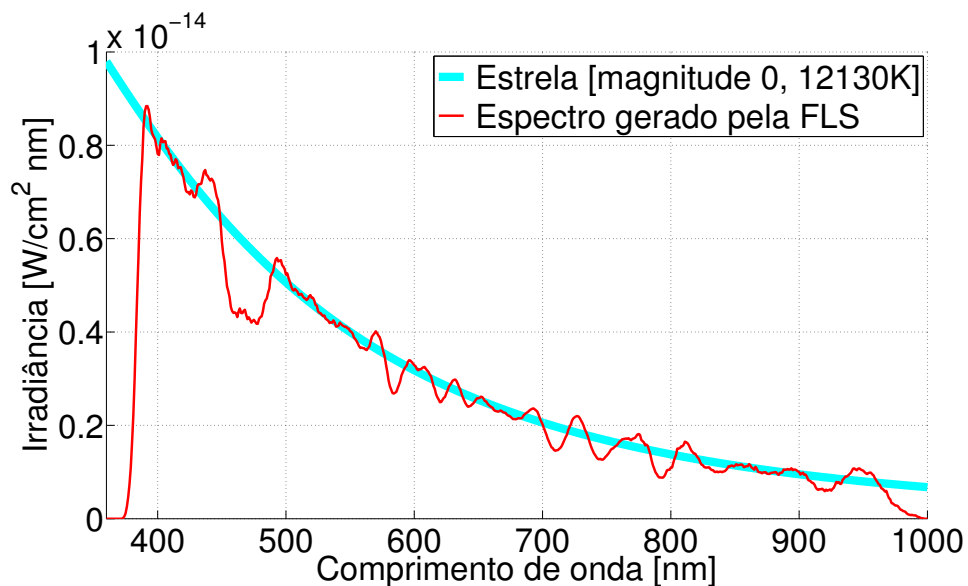


Figura 4.7 - Simulação de estrela de magnitude zero a 12 130K conseguido com GD.

4.3 Levenberg-Marquardt vs Gradiente Descendente

Para fins de comparar o número de iterações necessárias para convergência e o erro de ajuste de curva do LM e GD, os dois algoritmos foram implementados em MATLAB. O software desenvolvido para simular o comportamento da FLS apresentado no Capítulo 3.2 foi utilizado com uma pequena alteração: ao invés de usar os LEDs modelados, foi utilizado o banco de dados contendo as DPEs dos canais da FLS coletados anteriormente. Os testes foram realizados para a simulação de quatro estrelas de magnitude zero com temperaturas de 3 500K, 5 800K, 7 350K e 20 000K.

Os resultados dos testes foram conseguidos executando cada algoritmo 50 vezes. Em cada iteração foram gerados números aleatórios para definir a condição inicial dos algoritmos. O critério de parada foi definido como o número máximo de 500 iterações. A Figura 4.8 apresenta a comparação de convergência entre o algoritmo LM e GD para simular uma estrela de 3 500K. Os gráficos são apresentados no intervalo de 0 a 200 para o número de iterações, isso devido ao valor do ajuste de curva se apresentar quase constante a partir de 200 iterações.

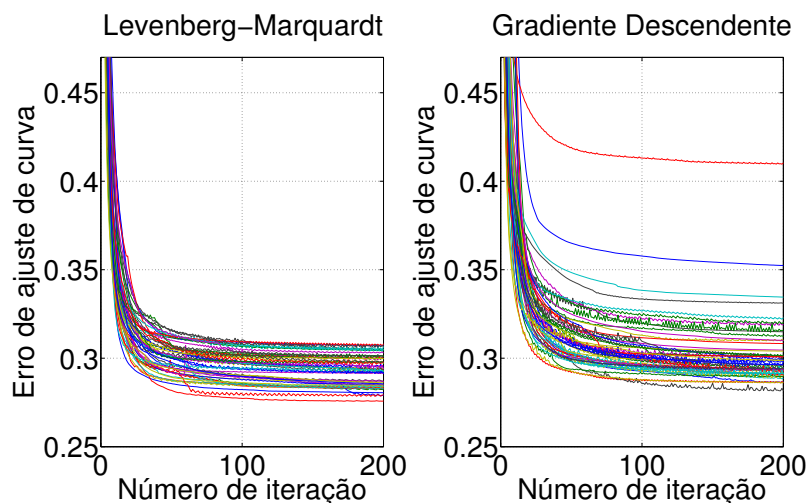


Figura 4.8 - Comparação da convergência dos algoritmos LM e GD, em 50 execuções independentes para simular uma estrela de 3 500K.

O LM apresentou convergência melhor que o GD. Embora em alguns casos ambos os algoritmos conseguiram erro de ajuste de curva próximo de 0,3, o GD se mostrou mais sensível aos valores definidos na condição inicial. A Figura 4.9 apresenta a comparação de convergência entre o algoritmo LM e GD para simular uma estrela de 5 800K.

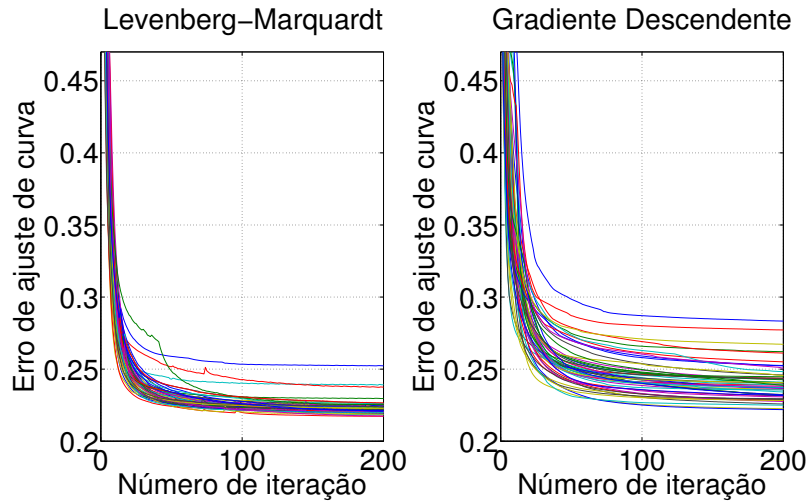


Figura 4.9 - Comparação da convergência dos algoritmos LM e GD, em 50 execuções independentes para simular uma estrela de 5 800K.

Analisando a convergência dos algoritmos para a estrela de 5 800K, percebe-se novamente que o GD é mais sensível a condição inicial do que o LM. Ao contrário do GD, a maior parte das simulações com LM apresentam boa convergência. A Figura 4.10 apresenta a comparação de convergência entre o algoritmo LM e GD para simular uma estrela de 7 300K. Cada uma das 50 execuções independentes foram realizadas com a condição inicial geradas aleatoriamente. Em todos os testes realizados, o LM demonstrou-se ser menos influenciado pela condição inicial do que o GD.

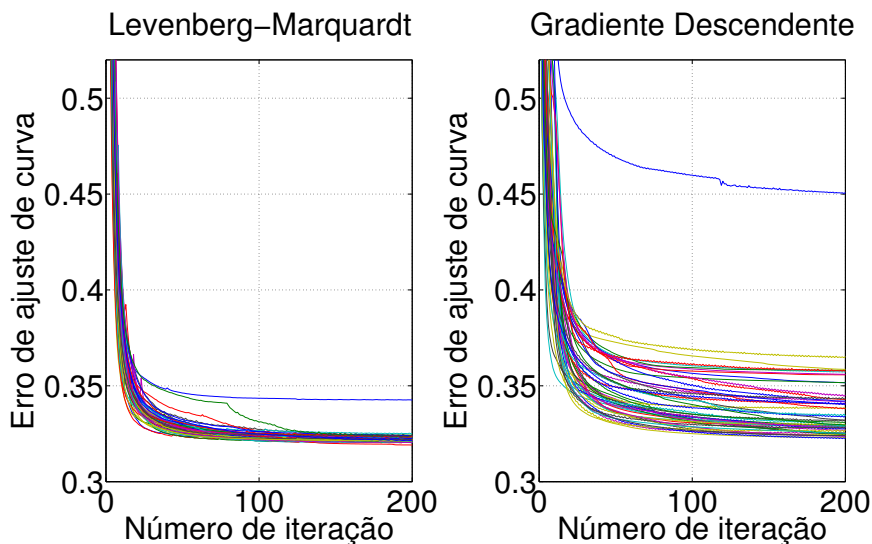


Figura 4.10 - Comparação da convergência dos algoritmos LM e GD, em 50 execuções independentes para simular uma estrela de 7 300K.

A Figura 4.11 apresenta a comparação de convergência entre o algoritmo LM e GD para simular uma estrela de 20 000K. O GD, algoritmo proposto por Fryc et al. (2004), Fryc et al. (2005a) e Fryc et al. (2005b) para ajuste de curva e simulação espectral se mostrou eficiente e com bons resultados, porém de convergência muito lenta.

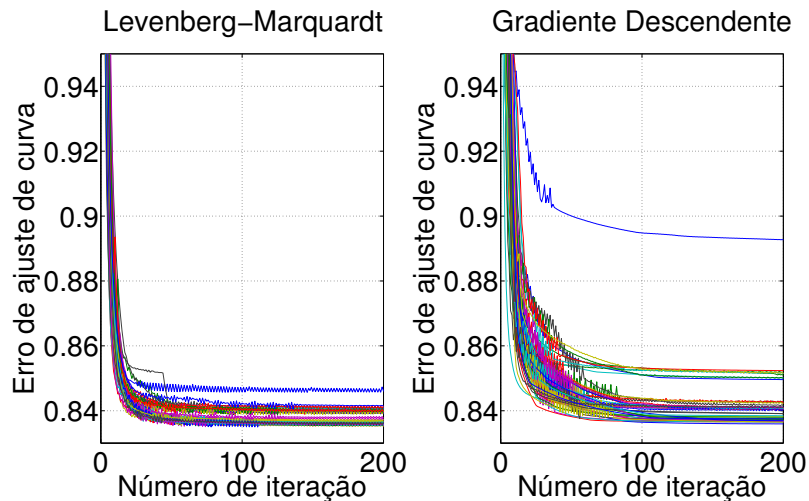


Figura 4.11 - Comparação da convergência dos algoritmos LM e GD, em 50 execuções independentes para simular uma estrela de 20 000K.

A vantagem que o GD tem sobre o LM é que não precisa da base dados com os valores das DPEs dos canais da FLS em função da potência. Com o tempo de convergência relativamente alto, o GD não se mostrou favorável para simulação de estrelas para a calibração do SEA. O LM se apresentou como o algoritmo mais rápido e no geral com melhores resultados.

4.4 Estabilidade da simulação das estrelas

Para a realização de testes e calibração do SEA, o espectro das estrelas geradas pelo simulador de estrelas deve se manter estável por vários dias. Sendo assim foi realizado uma avaliação de estabilidade do espectro gerado pela FLS em função do tempo.

Foi implementado e adicionado ao software da FLS uma função para coleta de dados com a possibilidade de definir o tempo de análise e o período de coleta. Uma estrela de magnitude zero e temperatura de 7 500K foi gerada usando o software da FLS e os dados coletados durante set dias. Os resultados demonstraram que o espectro gerado pela FLS decae com o tempo, conforme apresentam as Figuras 4.12 e 4.13.

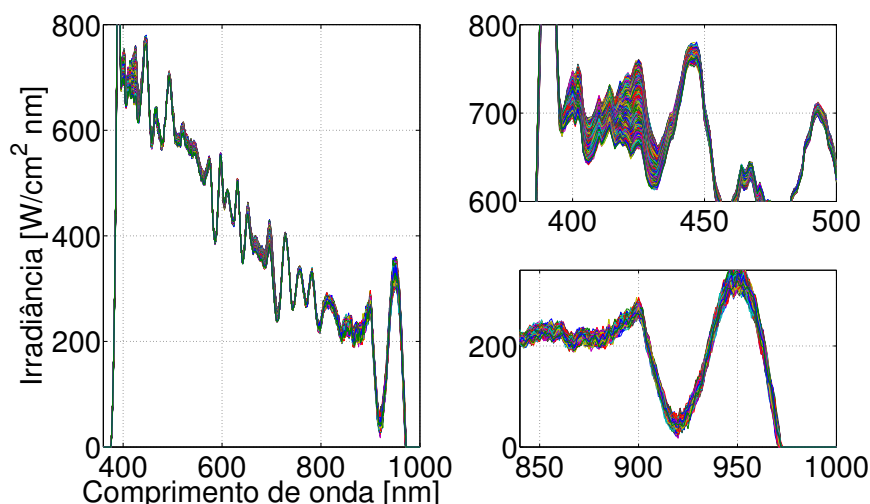


Figura 4.12 - Variação da irradiância espectral da FLS para uma estrela de magnitude zero a 7500K, com dados coletados durante sete dias.

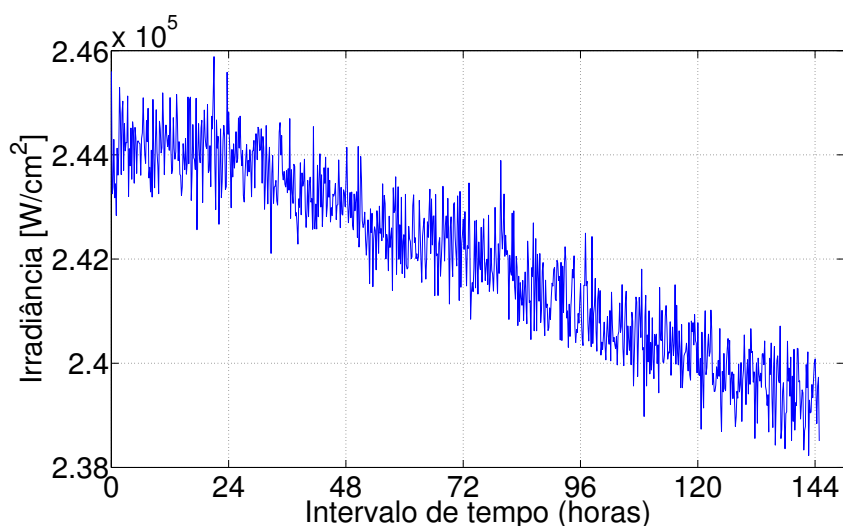


Figura 4.13 - Variação da irradiância da FLS para uma estrela de magnitude zero a 7500K, com dados coletados durante sete dias.

A estabilidade espectral é muito importante durante a caracterização e a calibração do sensor de estrelas, pois durante esse processo é analisando a influência do espectro contínuo da estrela no cálculo do erro da posição relativa da estrela detectada na matriz de pixels do detector do SEA. Segundo [Albuquerque \(2005\)](#), o conhecimento dos erros presentes na determinação da posição relativa de uma estrela é de extrema importância para que se possa caracterizar a precisão do SEA. O conhecimento da precisão é necessário para definir as tolerâncias no algoritmo de identificação de padrões do sensor de estrelas.

Essa variação no espectro contínuo das estrelas simuladas por um longo período de tempo gerou a necessidade de acrescentar ao software um sistema de controle que monitora e corrige alterações no espectro da FLS durante as simulações. Dessa forma, durante as simulações das estrelas, é usado o algoritmo LM ou GD para realizar o casamento espectral entre o espectro desejado e o obtido. Após a convergência do algoritmo, é registrado o espectro final gerado, e o sistema de controle da FLS entra em ação monitorando o oscilação espectral.

A cada 10 minutos é realizada uma leitura do espectro, e caso esse apresente alguma oscilação maior que 5% do espectro registrado, é acionado o sistema para correção do espectro de forma a garantir a mínima oscilação durante a simulação.

5 CONCLUSÃO

Neste trabalho foi proposto uma metodologia para definição de parâmetros de projeto de uma fonte de luz espectralmente sintonizável e o desenvolvimento de um sistema de controle usando métodos clássicos de otimização para simular estrelas. O processo todo envolveu estudos na área de radiometria, fotometria, engenharia espacial e computação aplicada.

Durante o projeto da FLS foi preciso definir de forma ótima a quantidade e tipos de LEDs necessários para gerar espectros de estrelas exigidos na calibração e testes do SEA. Procurou-se inicialmente resolver o problema utilizando métodos clássicos, todavia não se conseguiu resultados viáveis diante das dimensões e restrições de engenharia. Dessa forma, buscou-se uma solução com o algoritmo multiobjetivo M-GEO, que apresentou resultados satisfatórios e aceitáveis, conforme apresentado ao longo desse trabalho.

O GEO foi inspirado pela teoria da Criticalidade Auto-Organizada, e enquadra-se na categoria de métodos de otimização inspirados na natureza (bioinspirado), como o RS e o AG. Desta forma o GEO se classifica como um paradigma de Inteligência Artificial. Com relação ao RS e ao AG, o GEO e sua versão multiobjetivo, o M-GEO, têm a vantagem de possuírem apenas um parâmetro de ajuste, além das suas implementações serem muito simples. A performance de um algoritmo evolutivo em um dado problema é função do ajuste apropriado dos seus parâmetros livres que, em alguns casos, pode se tornar em mais um problema a ser solucionado. A parte mais complicada na implementação do algoritmo M-GEO foi o desenvolvimento da função que classifica o conjunto de soluções e gera a fronteira de Pareto.

Uma vez definido o conjunto de LEDs, diversas simulações foram realizadas para avaliar a performance da FLS. Como os resultados das simulações indicaram que a solução conseguida com o M-GEO atende os requisitos SEA, então o processo de compra dos LEDs foi encaminhado. Houve muita dificuldade para adquirir os LEDs selecionados, pois alguns fabricantes não comercializam pequenas quantidades e, em alguns casos, o andamento do processo de compra foi prejudicado por entraves burocráticos. Após os processos de definição dos parâmetros de projetos da FLS, do desenvolvimento de simuladores em computador, da aplicação destes simuladores para prever o comportamento da fonte de luz e da compra dos LEDs, a construção da FLS pôde ser concluída.

Um software em C++ usando a plataforma de desenvolvimento *open source* Qt foi implementado para controlar a FLS. Inicialmente foram implementados dois algoritmos clássicos de otimização, o LM e o GD, que se mostraram eficientes para simular estrelas. Diversos testes comparativos foram realizados entre o LM e o GD, e chegou-se a conclusão de que o LM é mais eficiente que o GD devido a sua velocidade de convergência e baixa sensibilidade à condição inicial.

Por fim foi realizado uma avaliação de estabilidade do espectro gerado pela FLS em função do tempo. Os resultados demonstraram que o espectro gerado pela FLS decaía com o tempo, o que gerou a necessidade de acrescentar ao software um sistema de controle que monitora e corrige alterações no espectro da FLS durante as simulações.

Há inúmeras atividades que requerem a correspondência espectral entre fontes de luz. Exemplo disso são as simulações da iluminação diurna para calibração de instrumentos ou inspeção visual. Utilizando LEDs com diferentes distribuição de potência espectral, e controlando a potência de saída de cada tipo de LED é possível ter uma fonte de luz multiespectral ajustável capaz de fornecer correspondência entre um espectro sintético e outro de referência com alta fidelidade. A FLS construída está sendo utilizada apenas para simular estrelas para a calibração do SEA, todavia espera-se também que ela possa ser utilizada em diversos outros estudos e simulações ópticas, não sendo limitada apenas a simulação de estrelas.

Alguns pontos que considera-se como objetos de desenvolvimentos de trabalhos futuros são:

- a) A definição dos parâmetros de projeto da FLS usando paradigmas de IA e algoritmos multiobjetivos ocorreu somente para a escolha ótima dos LEDs. Seria interessante melhorar a FLS usando um algoritmo evolutivo para definir não só a quantidade e tipos de LEDs, mas também definir o tamanho da esfera integradora de forma a conseguir os melhores ajustes de curvas.
- b) O algoritmo M-GEO implementado neste trabalho usa a versão canônica com representação das variáveis usando uma cadeia binária. Há outras versões do M-GEO usando codificação real. Será interessante a implementação do M-GEO real a fim de comparar com os resultados já conseguidos.
- c) O software implementado em C++ para controlar a FLS contém diversas funcionalidades, entre elas o controle individual da fonte de LEDs e do

espectrorradiômetro, a função para simular estrelas usando o LM ou o GD, e uma função para análise da estabilidade do espectro gerado na esfera integradora ao longo do tempo. Será de grande valia a análise de novos algoritmos para simular estrelas além do LM e do GD.

- d) Os valores para o fator de amortecimento, α , usado no LM e no GD foram definidos de forma empírica. Há na literatura diversos estudos para definição desse fator, e será valioso o estudo mais aprofundado para a definição ótima do α tanto para o LM quanto para o GD.

Finalmente, cabe observar que os resultados decorrentes da metodologia desenvolvida exposta ao longo dessa dissertação de mestrado, vêm sendo divulgados em reuniões e revistas técnico-científicas através de artigos e apresentações. O primeiro trabalho resultante do programa de mestrado foi um artigo ainda apenas sobre a proposta de desenvolvimento da FLS, que foi apresentado e publicado nos anais do I Simpósio de Ciência e Tecnologia do IEAv - I STCI (BORGES; GUIMARAES, 2012). Um segundo trabalho foi apresentado no Workshop dos cursos de Computação Aplicada do INPE, abordando a tentativa de solução do problema da escolha de LEDs usando um AG (BORGES et al., 2012). O terceiro trabalho abordando o algoritmo LM para simulação de estrelas foi apresentado no Workshop dos cursos de Computação Aplicada do INPE (BORGES; GUIMARAES, 2013b). Um quarto trabalho foi submetido para publicação na revista Journal of Computational Interdisciplinary Sciences (JCIS) (BORGES; GUIMARAES, 2013a), e apresenta a metodologia desenvolvida para a escolha ótima dos LEDs da FLS usando o algoritmo M-GEO.

REFERÊNCIAS BIBLIOGRÁFICAS

ABREU, B.; MARTINS, E.; DE SOUSA, F. L. Automatic test data generation for path testing using a new stochastic algorithm. In: SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE (SBES), 19., 2005, Uberlândia. **Proceedings...** Porto Alegre: SBC, 2005. p. 247–262. ISBN 85-7669-030-6. 22

ALBUQUERQUE, B. F. C. de. **Estudo dos erros sistemáticos inerentes a um sensor de estrelas de cabeça fixa, quando à localização relativa de estrelas.** 2005. 194 p. (INPE-14401-TDI/1127). Dissertação (Mestrado em Mecânica Espacial e Controle) — Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 2005. 1, 4, 6, 9, 11, 12, 60

ALLEN, C. W. **Astrophysical quantities.** 3rd. ed. London: Athlone Press, 1973. 310 p. 9, 10

BAK, P.; SNEPPEN, K. Punctuated equilibrium and criticality in a simple model of evolution. **Physical Review Letters**, v. 71, n. 24, p. 4083–4086, 1993. 22

BERTOLINO, M. A. **Proposta de Desenvolvimento de um Sensor de Estrelas Autônomo.** São José dos Campos: INPE, 2004. 12 p. Relatório Técnico Interno. 1, 5, 6, 9

BOETTCHER, S.; PERCUS, A. Optimization with extremal dynamics. **Physical Review Letters**, v. 86, p. 5211–5214, 2001. 22

BORGES, M. E. G.; ALBUQUERQUE, B. F. C.; GUIMARAES, L. N. F. Algoritmo genético para otimização de parâmetros de projeto de uma fonte de luz multiespectral baseada em LEDs. In: WORKSHOP DOS CURSOS DE COMPUTAÇÃO APLICADA DO INPE (WORCAP), 12., 2012, São José dos Campos. **Anais...** São José dos Campos: INPE, 2012. 65

BORGES, M. E. G.; GUIMARAES, L. N. F. Fonte multiespectral ajustável baseada em leds para geração de espectros sintéticos necessários na calibração de sensor estelar. In: SIMPÓSIO DE CIÊNCIA E TECNOLOGIA DO IEAV (STCI), 1., 2012, São José dos Campos. **Anais...** São José dos Campos: IEAv, 2012. ISSN 2238-5851. 65

_____. Generalized extremal optimization algorithm to design of a LED-based spectrally tunable light source for star simulation. **Journal of Computational Interdisciplinary Sciences (JCIS)**, JCIS, São José dos Campos, v. 4, 2013. 65

_____. Star simulation with LED-based spectrally tunable light source. In: WORKSHOP DOS CURSOS DE COMPUTAÇÃO APLICADA DO INPE (WORCAP), 13., 2013, São José dos Campos. **Anais...** São José dos Campos: INPE, 2013. 65

BRANKE, J.; DEB, K.; MIETTINEN, K.; SLOWIŃSKI, R. **Multiobjective optimization: interactive and evolutionary approaches**. Berlin: Springer-Verlag, 2008. 470 p. 20, 21

CARVALHO, G. B. **Levantamento de técnicas de identificação de estrelas e desenvolvimento de um ambiente de simulação e testes para análise de seus desempenhos em aplicações espaciais**. 2000. 286 p. (INPE-8307-TDI/766). Dissertação (Mestrado em Mecânica Espacial e Controle) — Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 2000. 4

CENTRE DE DONNÉS ASTRONOMIQUES DE STRASBOURG (CDS). **The Hipparcos and Tycho catalogues**. Paris, 2007. (CDS I/239). Disponível em: <<http://cdsarc.u-strasbg.fr/>>. Acesso em: 01 mar. 2012. 11

COMMISSION INTERNATIONALE DE L'ECLAIRAGE (CIE). **CIE 127:2007: measurement of leds**. 2nd. ed. Vienna, Austria, 2007. 40 p. 15, 18

DE SOUSA, F. L. **Otimização extrema generalizada: um novo algoritmo estocástico para o projeto ótimo**. 2002. 142 p. (INPE-9564-TDI/836). Tese (Doutorado em Computação Aplicada) — Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 2002. 22, 23, 24, 25

DE SOUSA, F. L.; RAMOS, F. M.; PAGLIONE, P.; GIRARDI, R. M. New stochastic algorithm for design optimization. **American Institute of Aeronautics and Astronautics**, AIAA, v. 41, n. 9, p. 1808–1818, 2003. 22, 38

DE SOUSA, F. L.; VLASSOV, V.; RAMOS, F. M. Generalized extremal optimization: An application in heat pipe design. **Applied Mathematical Modelling**, Elsevier, v. 28, n. 10, p. 911–931, 2004. ISSN 0307-904X. 22

EIBEN, A. E.; SMITH, J. E. **Introduction to evolutionary computing**. Berlin: Springer, 2007. 300 p. 20, 21

EUROPEAN SPACE AGENCY (ESA). **The Hipparcos and Tycho catalogues**. Paris, 1997. (ESA SP-1200). Disponível em: <<http://www.rssd.esa.int/Hipparcos/>>. Acesso em: 01 mar. 2012. 11, 12

FIALHO, M. A. A. **Estudo Comparativo entre dois algoritmos de identificação de estrelas para um sensor de estrelas autônomo de campo largo**. 2007. 237 p. Dissertação (Mestrado em Engenharia Eletrônica e Computação) — Instituto Tecnológico de Aeronáutica (ITA), São José dos Campos, 2007. [1](#), [4](#), [6](#), [9](#)

FRYC, I.; BROWN, S. W.; EPPELDAUER, G. P.; OHNO, Y. A spectrally tunable solid-state source for radiometric, photometric, and colorimetric applications. In: INTERNATIONAL CONFERENCE ON SOLID STATE LIGHTING, 4., 2004, Denver. **Proceedings...** Denver, CO, United States: Proc. SPIE, 2004. v. 5530, p. 150–159. [54](#), [59](#)

_____. LED-based spectrally tunable source for radiometric, photometric, and colorimetric applications. **Optical Engineering**, SPIE, v. 44, n. 11, p. 111309, 2005. [14](#), [54](#), [59](#)

FRYC, I.; BROWN, S. W.; OHNO, Y. Spectral matching with an LED-based spectrally tunable light source. In: INTERNATIONAL CONFERENCE ON SOLID STATE LIGHTING, 5., 2005, San Diego. **Proceedings...** San Diego, CA, USA: Proc. SPIE, 2005. v. 5941, p. 59411I–59411I–9. [15](#), [54](#), [59](#)

GALSKI, R. L. **Desenvolvimento de versões aprimoradas híbridas, paralela e multiobjetivo do método da otimização extrema generalizada e sua aplicação no projeto de sistemas espaciais**. 2006. 279 p. (INPE-14795-TDI/1238). Tese (Doutorado em Computação Aplicada) — Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 2006. [22](#), [23](#), [24](#), [25](#), [26](#), [27](#), [28](#), [38](#)

HALLIDAY, D.; RESNICK, R.; KRANE, K. S. **Física 4**. 4. ed. Rio de Janeiro: LTC Livros Técnicos e Científicos Editora S.A., 1996. 338 p. [10](#)

LABSPHERE. **A Guide to Integrating Sphere Radiometry and Photometry**. US, 2012. [14](#)

LEVENBERG, K. A method for the solution of certain non-linear problems in least-squares. **Quarterly Journal of Applied Mathematics**, v. 2, n. 2, p. 164–168, 1944. [34](#)

LOPES, I. M. L.; DE SOUSA, F. L.; SOUZA, L. C. G. de. The generalized extremal optimization with real codification. In: INTERNATIONAL CONFERENCE ON ENGINEERING OPTIMIZATION (ENGOPT), 1., 2008, Rio

de Janeiro. **Proceedings...** Rio de Janeiro: EngOpt, 2008. ISBN 978-85-7650-344-6. 22, 25

MARQUARDT, D. W. An algorithm for least-squares estimation of nonlinear parameters. **Society for Industrial and Applied Mathematics**, v. 11, n. 2, p. 431–441, 1963. 34

MATOS, J. D. **Dimensionamento Radiométrico Preliminar do Sensor de Estrelas**. São José dos Campos: INPE, 1997. 22 p. Relatório Técnico Interno. 4

MEJÍA CABEZA, J. **Um sensor estelar para o apontamento fino do telescópio MASCO**. 1997. 99 p. (INPE-6682-TDI/627). Dissertação (Mestrado em Ciência Espacial) — Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 1997. 3

OHNO, Y. **Photometric calibrations, NIST special publication 250-37**. Gaithersburg, U.S., 1997. 85 p. Disponível em: <www.nist.gov/calibrations>. 18

_____. Color rendering and luminous efficacy of white led spectra. In: INTERNATIONAL CONFERENCE ON SOLID STATE LIGHTING, 4., 2004, Denver. **Proceedings...** Denver, CO, United States: Proc. SPIE, 2004. v. 5530, p. 88–98. 15

_____. Spectral design considerations for white led color rendering. **Optical Engineering**, SPIE, v. 44, n. 11, p. 111302, 2005. 15

OPTICAL SOCIETY OF AMERICA. **Handbook of optics**. US: McGraw-Hill Inc., 1978. 1172 p. ISBN 978-00-704-7710-0. 13, 14

OZISIK, M. N.; ORLANDE, H. R. B. **Inverse Heat Transfer: fundamentals and applications**. USA: Taylor & Francis, 2000. 352 p. 34

RADIANT ZEMAX LLC. **Zemax optical design program: user's manual**. Tucson, 2011. 16

REED, B. C. The composite observational-theoretical HR diagram. **Journal of the Royal Astronomical Society of Canada**, v. 1, n. 92, p. 36–37, 1998. 11

SCHUBERT, E. F. **Light-emitting diodes**. 2nd. ed. UK: Cambridge University Press, 2006. 431 p. ISBN 9780521865388. 10, 15, 17

SMITH, W. J. **Modern optical engineering**: the design of optical systems. USA: McGraw-Hill, 1996. 476 p. 10, 13, 14

_____. **Modern optical engineering**. USA: McGraw-Hill, 2007. 764 p. 10, 13, 14

SWITALSKI, P.; SEREDYNSKI, F. Generalized extremal optimization for solving multiprocessor task scheduling problem. **Simulated Evol. and Learning Lecture Notes in Comp. Science**, v. 5361, p. 161–169, 2008. 22

_____. A grid scheduling based on generalized extremal optimization for parallel job model. In: INTERNATIONAL CONFERENCE ON PARALLEL PROCESSING AND APPLIED MATHEMATICS, 9., 2011, Berlin. **Proceedings...** Berlin, Heidelberg: Springer-Verlag, 2011. Part II (PPAM'11), p. 41–50. 22

_____. An effective multiprocessor scheduling with use of geo metaheuristic. **Int. J. Found. Comput. Sci.**, v. 23, p. 465–481, 2012. 22

VANDERPLAATS, G. N. **Numerical optimization techniques for engineering design**. 2nd. ed. Colorado Springs: Vanderplaats Research & Development, 1998. 338 p. 20

WALL, C. F.; HANSON, A. R.; TAYLOR, J. A. F. **Construction of a programmable light source for use as a display calibration artifact**. 2001. 259-266 p. 14

WERTZ, J. R. **Spacecraft attitude determination and control**. Torrance, CA, USA: Microcosm Inc, 1978. 858 p. ISBN 978-90-277-1204-2. 1, 5

WERTZ, J. R.; LARSON, W. J. **Space mission analysis and design**. 3rd. ed. Hawthorne, CA 90250, USA: Microcosm Press, 1999. 969 p. ISBN 978-0-7923-5901-2. 1, 5, 6

APÊNDICE A - LEDs candidatos a serem utilizados na FLS

A tabela abaixo apresenta os 160 LEDs candidatos a serem utilizados na FLS. O cabeçalho da tabela utiliza as seguintes abreviações:

- I_v : Intensidade Luminosa (mcd).
- θ_{fwhm} : Ângulo de visão (graus).
- λ_0 : Comprimento de onda de pico (nm).
- $FWHM$: Largura de meia potência (nm).
- Φ_e : Potência Radiante (mW).
- I_e : Intensidade Radiante (mW/sr).

Tabela A.1 - LEDs candidatos

LED	I_v	θ_{fwhm}	λ_0	FWHM	Φ_e	I_e
RLT360-1.0-15		30	361	10	0,75	
L375R-04		38	375	10	2,5	3,5
LED375-01	25	12	375	17	2,5	10
LED375-02		6	375	17	2,5	
LED375-03	30	14	375	17	2,5	8
LED375-04	15	34	375	17	2,5	4
LED375-05	3	92	375	17	2,5	1
LED375-06	70	6	375	17	2,5	30
LED375-33	15	22	375	17	2,5	7
LED375-36	7	46	375	17	2,5	3
RLS-UV380		30	380	17	9	
LED385-01	20	12	385	17	3,5	20
LED385-02	25	6	385	17	3,5	25
LED385-03	18	14	385	17	3,5	15
LED385-04	6	34	385	17	3,5	6
LED385-05	2	92	385	17	3,5	1
LED385-06	30	6	385	17	3,5	25
LED385-33	8	22	385	17	3,5	10
LED385-36	4	46	385	17	3,5	4
* LED395-01V	100	20	395	15	6	
LED395-02V	130	10	395	15	6	

(Continua)

Tabela A.1 - Continuação

LED	I_v	θ_{fwhm}	λ_0	FWHM	Φ_e	I_e
LED395-03V	90	24	395	15	6	
LED395-04V	35	46	395	15	6	
LED395-05V	15	60	395	15	6	
LED395-06V	140	18	395	15	6	
LED395-33V	45	36	395	15	6	
LED395-36V	20	56	395	15	6	
* GB-333UV	290	25	405	16		
LED405-01V	40	20	405	15	6	
LED405-02V	50	10	405	15	6	
LED405-03V	25	24	405	15	6	
LED405-04V	11	46	405	15	6	
LED405-05V	2	60	405	15	6	
LED405-06V	60	18	405	15	6	
LED405-33V	25	36	405	15	6	
LED405-36V	10	56	405	15	6	
* VL415-5-15		15	415	45	12	
L420R-03		20	420	15	15	40
* LED425-6-30		30	425	16	6	
383-UBC	1250	12	430	65	0,25	
L430-36U	75	60	430	50	0,5	
LED430-01U	300	20	430	50	0,3	
LED430-03U	150	30	430	50	0,3	
LED430-04U	80	40	430	50	0,3	
LED430-06U	600	12	430	50	0,3	
L435-06	2000	14	435	25	4,5	
* LED435-03	750	24	435	25	4,5	
LED450-01U	250	20	450	30	1,3	
LED450-02U	600	10	450	30	1,3	
LED450-03U	250	24	450	30	1,3	
* LED450-06U	1000	18	450	30	1,3	
LED450-36U	50	56	450	30	1,3	
5B4HCA-H	3000	36	460	30		
B5B-437-IX	3800	8	468	30		
* HLMP-CB31	1500	30	470	35		
L470-03V	2000	24	470	25	5	
* LED490-03U	1400	24	490	30	1,2	

(Continua)

Tabela A.1 - Continuação

LED	I_v	θ_{fwhm}	λ_0	FWHM	Φ_e	I_e
383-2UBGC	2500	6	502	30		
ELD-505-524	7000	15	505	30		
L505-01	17000	16	505	35	8	
L505-02	20000	10	505	35	8	
L505-03	17000	20	505	35	8	
L505-03	17000	20	505	35	8	
L505-04	5000	40	505	35	8	
L505-05	850	100	505	35	8	
* L505-06	24000	8	505	35	8	
B3B-443-B505	4000	15	507	30		
HLMP-CM31	4200	30	524	47		
LED525-01		18	525	35	7	35
* B5-433-B525	10000	15	528	30		
LED535-01	9500	16	535	35	4	14
LED535-02	10000	10	535	35	4	16
LED535-03	10000	20	535	35	4	15
LED535-04	3800	40	535	35	4	6
L545-01	20000	16	545	40	8	25
L545-02	21000	10	545	40	8	27
L545-03	21000	20	545	40	8	27
* L545-04	8000	40	545	40	8	12
L545-05	1400	100	545	40	8	1,2
L545-06	36000	8	545	40	8	40
LED545-01	20000	16	545	40	8	25
LED545-02	21000	10	545	40	8	27
LED545-03	21000	20	545	40	8	27
LED545-04	8000	40	545	40	8	12
TLPGE23TP	3000	2	562	14		
L565-01	600	20	565	25	0,2	
L565-03	400	30	565	25	0,2	
L565-04	300	40	565	25	0,2	
L565-04U	300	40	565	25	0,2	
* L565-06	1000	12	565	25	0,2	
TLFGE23TP	5000	2	568	15		
TLGE23TP	7000	2	574	17		
* B5-433-20D	1700	30	575	25		

(Continua)

Tabela A.1 - Continuação

LED	I_v	θ_{fwhm}	λ_0	FWHM	Φ_e	I_e
TLPYE23TP	8000	2	583	14		
L590-33W	1700	36	590	20	2,5	
* OPE5T59UY	2000	16	591	15		
L600-33V	1300	36	600	15	1	
* LED600-03V	2000	22	600	15	1	
B5-435-30S	4000	15	610	20		
* B5B-435-30S	6000	8	610	20		
* B5-435-30	4000	15	625	20		
B5-435-30	4000	15	625	20		
* B5-435-TL	10000	15	630	20		
L630-03W	6000	20	630	20	9	
HLMP-EG32	1500	30	635	17		
* ELD-650-523	500	20	650	17,3	1,4	5
* XR65-R5P0U		18	650	7	2,5	
* B5B-436-30	3500	8	660	25		
* ELD-670-524	780	20	670	25	8	
L680-01AU		20	680	25	4	25
* L680-02AU		14	680	25	4	25
L680-06AU		14	680	25	4	25
* ELD-700-524		20	700	45	10	
L700-01AU		20	700	30	13	60
L720-01AU		20	720	30	13	60
L720-06AU		14	720	30	13	80
* LED740-01AU		20	740	30	18	90
L750-03AU		20	750	30	19	100
L760-01AU		20	760	30	19	100
* L760-03AU		20	760	30	19	100
* ELD-770-524		20	770	30	10	70
ELD-780-524		20	780	25	8,5	35
L78-01AU		20	780	30	20	100
* L780-06AU		14	780	30	20	150
L800-01AU		20	800	35	22	100
* ELD-810-525		20	810	35	28	
* LED810-03AU		20	810	35	22	100
L820-01AU		20	820	35	18	100
L820-02AU		14	820	35	18	100

(Continua)

Tabela A.1 - Continuação

LED	I_v	θ_{fwhm}	λ_0	FWHM	Φ_e	I_e
L820-03AU		20	820	35	18	110
L820-04AU		40	820	35	18	45
L820-36AU		66	820	35	18	20
* LED830-03AU		30	830	35	18	80
* EL-HIR204		25	850	45		90
FL850-03-80		16	850	40	35	230
* LED870-01UP		20	870	35	22	90
* ELD-880-525		20	880	25	32	
LED890-01AU		20	880	75	15	120
LED890-02AU		10	880	75	15	140
* LED890-03AU		30	880	75	15	100
LED890-04AU		40	880	75	15	60
LED890-33AU		30	880	75	15	40
LED890-36AU		60	880	75	15	30
L890-01AU		20	890	50	15	70
L890-06AU		14	890	50	15	100
L890-41AU		32	890	50	15	45
L890-42AU		46	890	50	15	25
B5B-900-8		8	900	80		48
L910-01AU		20	910	45	12	70
* L910-06		14	910	45	12	65
ELD-920-535		30	920	65	29	
ELD-935-525		20	935	65	30	100
B5B-940-8		8	940	50		48
* OPE5594S		10	940	45		10
* ELD-950-535		26	950	65	26	75
ELD-950-545-2		40	950	50	26	45
ELD-960-545		40	960	60	27	45
L970-01		20	970	55	5,5	25
* L970-06		14	970	55	5,5	25
LED980-03		22	985	45	4	10

Fonte: Autor.

* LEDs selecionados.

APÊNDICE B - Rotinas em MATLAB

Rotina B.1 - Cálculo da distribuição de potência espectral do LED

```
1 function s_led = spd_led(lambda, lambda0, dlambda05)
2
3 % function s_led = spd_led(lambda, lambda0, dlambda05)
4 %
5 % Spectral Power Distribution (SPD) of a model LED.
6 %
7 % REFERENCES:
8 % OHNO, Y. Spectral design considerations for white LED color rendering
9 % DOI = 10.1117/1.2130694
10 %
11 % SCHUBERT, E. F. Light-emitting diodes. 2nd. ed.
12 % UK: Cambridge University Press, 2006. 431 p. ISBN 9780521865388.
13 %
14 % COMMISSION INTERNATIONALE DE L'ECLAIRAGE (CIE). CIE 127:2007:
15 % Measurement of leds. 2nd. ed. Vienna, Austria, 2007. 40 p.
16 % -----
17 %
18 % Input:
19 % lambda : Wavelength in nanometer
20 % lambda0 : Peak wavelength
21 % dlambda05: Delta Lambda 1/2, Spectrum Radiation Bandwidth, FWHM spectral
22 %
23 % Output:
24 % s_led(:, 1): wavelength
25 % s_led(:, 2): normalized LED Spectral Power Distribution
26 %
27 % Rev 0, August 2012 (Marcos Eduardo Borges | marcosedborges@gmail.com)
28
29 gaussian_led = exp(-((lambda - lambda0) / dlambda05).^2);
30 s_led(:, 1) = lambda;
31 s_led(:, 2) = (gaussian_led + 2 * (gaussian_led.^5)) / 3;
```

Rotina B.2 - Cálculo da distribuição de intensidade luminosa do LED por ângulo

```
1 function i_led = ipd_led(view_angle_deg, intensity0)
2
3 % function i_led = ipd_led(view_angle_deg, intensity0)
4 %
5 % Intensity Power Distribution by Angle
6 %
7 % REFERENCE:
8 % RADIANT ZEMAX LLC. Zemax optical design program: user's manual. Tucson, 2011.
9 % -----
10 %
11 % Input:
12 % view_angle_deg: View Angle, 2*theta_1/2, Angle for half Intensity (FWHM)
13 % intensity0 : Intensity value in angle = 0
14 % intensity0 = 1 for normalized intensity.
15 %
16 % Output:
17 % i_led(:, 1): angle theta
18 % i_led(:, 2): LED Intensity power distribution by angle
19 %
20 % Rev 0, August 2012 (Marcos Eduardo Borges | marcosedborges@gmail.com)
21
22 alpha_x = view_angle_deg / sqrt(2 * log(2));
23 G_x = 1;
24 theta_x = 0:1:180;
25 i_led(:, 1) = theta_x;
26 i_led(:, 2) = intensity0 * exp(-2 * (theta_x / alpha_x).^2 * G_x);
```

Rotina B.3 - Cálculo do fluxo do LED

```

1 function flux = flux_led(theta_deg, intensity_by_angle, maxIntensity)
2
3 % function flux = flux_led(theta_deg, intensity_by_angle, maxIntensity)
4 %
5 % Calculating Flux (Lumen or Watt)
6 % The Flux result in 'flux' will be in units of 'maxIntensity'
7 %
8 % REFERENCES:
9 % SCHUBERT, E. F. Light-emitting diodes. 2nd. ed.
10 % UK: Cambridge University Press, 2006. 431 p. ISBN 9780521865388.
11 %
12 % COMMISSION INTERNATIONALE DE L'ECLAIRAGE (CIE). CIE 127:2007:
13 % Measurement of leds. 2nd. ed. Vienna, Austria, 2007. 40 p.
14 % -----
15 %
16 % Input:
17 %   theta_deg           : Angles in degrees
18 %   intensity_by_angle : Intensity by angle (Normalized or Non-normalized)
19 %   maxIntensity        : Luminous Intensity (Candela), Radiant Intensity (Watt/sr)
20 %
21 % Output:
22 %   flux: Luminous Flux (Lumen) if maxIntensity in Candela (Lumen/sr)
23 %         Radiant Power (Watt) if maxIntensity in Watt/sr
24 %
25 % Rev 0, August 2012 (Marcos Eduardo Borges | marcoseborges@gmail.com)
26
27 % Ensuring proper data struct
28 if size(theta_deg,1) == 1
29     theta_deg = transpose(theta_deg);
30 end
31
32 if size(intensity_by_angle,1) == 1
33     intensity_by_angle = transpose(intensity_by_angle);
34 end
35
36 % Angle must be in radians
37 % solid_angle = 2 * pi * sum{i=1..N-1}[(f_i + f_{i-1}) / 2 * ΔTheta]
38 f_i = intensity_by_angle .* sin( theta_deg * (pi/180) );
39 ΔTheta = (pi/180) * (theta_deg( 2:size(theta_deg,1) ) - theta_deg( ...
40     1:size(theta_deg,1)-1 ));
41 solid_angle = 2 * pi * sum( (f_i(2:size(f_i,1)) + f_i(1:size(f_i,1)-1)) / 2 .* ΔTheta );
42 flux = solid_angle * maxIntensity;

```

Rotina B.4 - Cálculo do fator de conversão F_{LED}

```

1 function F_led = f_led(s_led, V, inputType, inputValue)
2
3 % function conversionFactor = f_led( s_led, V, inputType, inputValue )
4 %
5 % Conversion Factor of Normalized Luminous Intensity to Power in Watt
6 %
7 % REFERENCES:
8 % OHNO, Y. Spectral design considerations for white LED color rendering
9 % DOI = 10.1117/1.2130694
10 %
11 % SCHUBERT, E. F. Light-emitting diodes. 2nd. ed.
12 % UK: Cambridge University Press, 2006. 431 p. ISBN 9780521865388.
13 %
14 % COMMISSION INTERNATIONALE DE L'ECLAIRAGE (CIE). CIE 127:2007:
15 % Measurement of leds. 2nd. ed. Vienna, Austria, 2007. 40 p.
16 % -----
17 %
18 % Input:
19 %   s_led           : LED Spectral Power Distribution (SPD)
20 %   V               : Human Eye Response
21 %   inputType      : [1] Radiant Power (Watt)
22 %                   [2] Luminous Flux (Lumens)
23 %   inputValue     : Value of Type Specified in the variable inputType
24 %
25 % Output:
26 %   F_led: Conversion Factor
27 %

```

```

28 % Rev 0, August 2012 (Marcos Eduardo Borges | marcoseborges@gmail.com)
29
30 switch inputType
31     case 1 % Radiant Power (Watt)
32         F_led = inputValue / trapz(s_led(:,1), s_led(:,2));
33
34     case 2 % Luminous Flux (Lumens)
35         F_led = inputValue / (683 * trapz(s_led(:,1), s_led(:,2) .* V(:,2)));
36
37     otherwise
38         disp('inputType not defined in F_led = f_led()');
39 end

```

Rotina B.5 - Cálculo do fator de conversão esfera integradora, pinhole e colimador

```

1 function transferFunction = transfer_function()
2
3 % transferFunction = transfer_function()
4 %
5 % Calculating Transfer Function of the spectrum generated in the
6 % integrating sphere passing through the pinhole and collimator
7 %
8 % REFERENCE:
9 % LABSPHERE. A Guide to Integrating Sphere Radiometry and Photometry. US, 2012.
10 % -----
11 %
12 % Input: none
13 %
14 % Output:
15 % Transfer Function Factor: Irradiance W/cm^2.nm
16 %
17 % Rev 0, August 2012 (Marcos Eduardo Borges | marcoseborges@gmail.com)
18
19 % Radiance (Integrating Sphere Output) = L (W/cm^2 sr nm)
20 integratingSphereInputFlux = 1;
21 integratingSphereDiameter = 20;
22 integratingSphereInputDiameter = 2.50;
23 integratingSphereOutputDiameter = 2.50;
24 integratingSphereReflectanceFactor = 0.97;
25
26 p = integratingSphereReflectanceFactor;
27 phi = integratingSphereInputFlux;
28
29 As = 4 * pi * (integratingSphereDiameter / 2)^2; % Integrating Sphere Area
30 Ai = pi * (integratingSphereInputDiameter / 2)^2; % Integrating Sphere Input Area
31 Ao = pi * (integratingSphereOutputDiameter / 2)^2; % Integrating Sphere Output Area
32
33 L = (phi / (pi * As)) * (p / ((1 - p)*(1 - ((Ai + Ao) / As))));
34
35 % Irradiance (Collimator Output) = E (W/cm^2/nm)
36 pinholeDiameter = 5e-3;
37 beamDiameterOfTheCollimator = 20;
38 fNumberOfTheCollimator = 7.5;
39
40 Ap = pi * (pinholeDiameter / 2)^2; % Pinhole Area
41 Ac = pi * (beamDiameterOfTheCollimator / 2)^2; % Beam Area of the Collimator
42
43 E = 2 * pi * L * Ap * (1 / (8 * fNumberOfTheCollimator^2 * Ac));
44
45 transferFunction = E;

```

Rotina B.6 - Irradiância espectral da estrela

```

1  function starIrradiance = star_irradiance(wavelength, magnitude, temperature)
2
3  % function starSpectralIrradiance = star(wavelength, magnitude, temperature)
4  %
5  % Spectral Power Distribution of a model Star (W cm-2 nm-1)
6  %
7  % REFERENCES:
8  % Allen, C. W. Astrophysical Quantities. Great Britain: Willian Clowes,
9  % 1973. 197-209.
10 %
11 % HALLIDAY, D.; RESNICK, R.; KRANE, K. S. Física 4. 4. ed.
12 % Rio de Janeiro: LTC Livros Técnicos e Científicos Editora S.A., 1996. 338 p.
13 %
14 % Matos, José Dias. Dimensionamento radiométrico preliminar do sensor de estrelas.
15 % São José dos Campos: INPE, 1997. Relatório Técnico.
16 % -----
17 %
18 % Input:
19 %   wavelength : Wavelength in nanometers
20 %   magnitude   : Visual Magnitude of the Star
21 %   temperature: Temperature in Kelvin
22 %
23 % Output:
24 %   starIrradiance(:, 1) = wavelength
25 %   starIrradiance(:, 2) = Star modeled by the irradiance of a blackbody
26 %
27 % Rev 1, October 2013 (Marcos Eduardo Borges | marcoseborges@gmail.com)
28
29 % Important constants
30 % The NIST Reference on Constants, Units, and Uncertainty
31 % http://physics.nist.gov/cuu/Constants/index.html
32 k = 1.3806488e-23; % Boltzmann's constant J/K
33 h = 6.62606957e-34; % Planck's constant Js
34 c = 299792458; % Speed of light m/s
35
36 % Converts Wavelengths in nanometers to meters
37 WL = wavelength .* 1e-9;
38
39 % Blackbody in W m-2 m-1
40 blackbody = (2*pi*h*c^2) ./ (WL.^5.*(exp((h*c) ./ (k.* WL.*temperature)) - 1));
41
42 % Converts (m-2 m-1) to (cm-2 nm-1)
43 blackbody = blackbody ./ 1e13;
44
45 % Calculates Apparent Magnitude Transference Function (See Allen 1973)
46 visualMagnitude = 10^(-0.4 * magnitude) * 4e-15;
47
48 % Finds irradiance value at 550nm
49 index550 = find(wavelength & wavelength == 550);
50 normalizationFactor = visualMagnitude / blackbody(index550);
51
52 % Spectral Irradiance of a model Star (W cm-2 nm-1)
53 starIrradiance(:, 1) = wavelength;
54 starIrradiance(:, 2) = blackbody * normalizationFactor;

```

APÊNDICE C - Manual do Software do Simulador de Estrelas

O software do Simulador de Estrelas foi desenvolvido em C++ usando a plataforma de desenvolvimento Qt 5, e roda em máquinas Windows XP e Windows 7 com arquiteturas x86 e x64. Para comunicação, controle e aquisição de dados por meio do espectrorradiômetro SMS-500 é necessário instalar o software fornecido pela SphereOptics, empresa fabricante do espectrorradiômetro.

Para comunicação e controle da Fonte de Luz Espectralmente Sintonizável por meio da USB é necessário a instalação do driver do chip da FTDI, disponível para download em <http://www.ftdichip.com/Drivers/D2XX.htm>.

Ao executar o software pela primeira vez, será exibido a tela apresentada na Figura D.1. Selecione as opções conforme demonstrado na figura para concluir a instalação corretamente.

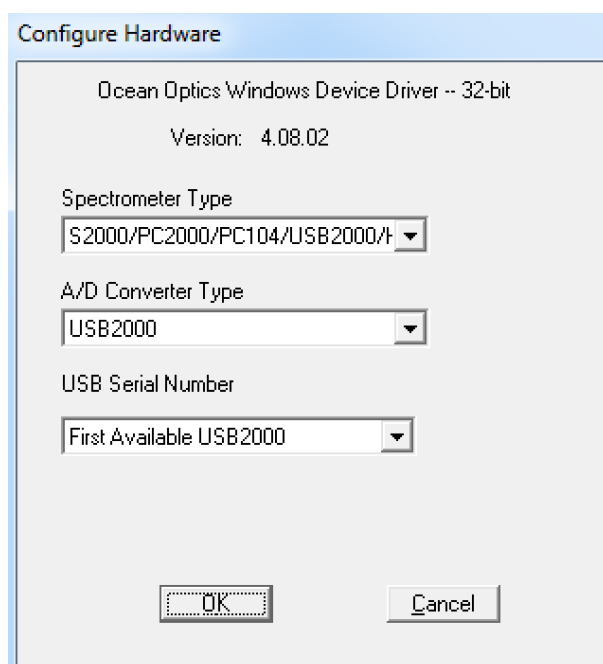


Figura C.1 - Configuração inicial do SMS-500.

C.1 Instalação e configuração do software SphereOptics SMS-500

Todos os arquivos necessários para instalação do SMS-500 estão disponíveis para download em <http://goo.gl/Yk2nML>. Abaixo é apresentado o processo de instalação:

1. Baixe e descompacte o arquivo **Labsphere SMS500.iso**.
2. Utilize o arquivo **SETUP.exe** para realizar a instalação.
3. Baixe e instale o driver para Windows 7 disponível no arquivo **SMS500 driver Win7x86.zip** ou **SMS500 driver Win7x64.zip** conforme a arquitetura do sistema operacional.

C.2 Módulo SMS500

O módulo SMS500 pode ser utilizado para controlar de forma independente o espectrorradiômetro SMS-500. Com esse módulo é possível realizar medições de fluxo, intensidade, irradiância ou radiância. A escolha do modo de operação pode ser realizada por meio do menu **Modes of Operation**. A Figura C.2 apresenta o módulo SMS500 do Software do Simulador de Estrelas.

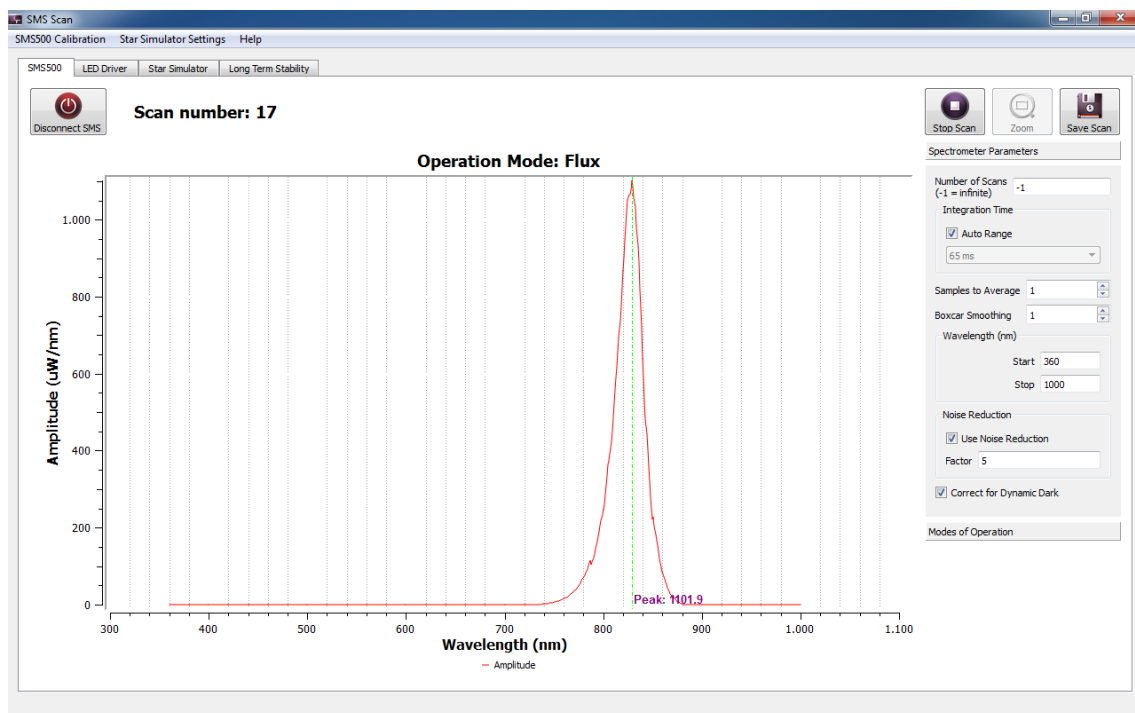


Figura C.2 - Software do Simulador de Estrelas: Módulo SMS500.

C.2.1 Procedimento de calibração do SMS-500

O espectrorradiômetro SMS-500 já vem calibrado de fábrica e com os arquivos de calibração apropriados. Existe porém a necessidade de calibrar o equipamento para realização de medições específicas, e também, para a definição do ponto zero para máxima acurácia. Existem dois tipos de arquivos de calibração associados ao software, ***.cal** e ***.dat**. Os arquivos ***.cal** contêm informações de fontes de calibração em $\mu\text{W}/\text{nm}$, $\mu\text{W}/\text{cm}^2 \text{ nm}$ ou $\mu\text{W}/\text{cm}^2 \text{ sr nm}$ por comprimento de onda. Os arquivos ***.dat** são arquivos intermediários de calibração interna do software.

Existem três arquivos ***.dat** principais:

1. “Flux.dat”: usado para medições do fluxo.
2. “Irradiance.dat”: usado para medições da irradiância ou intensidade.
3. “Radiance.dat”: usado para medições da radiância.

A escolha do modo de operação no software, ativa de forma automática o arquivo ***.dat** correspondente ao modo escolhido. A seleção de um modo de operação para o qual não há arquivo ***.dat** correspondente irá gerar resultados de medições sem sentido. Por exemplo, se houver apenas o arquivo de calibração para a radiância, e for tentado medir irradiância, os resultados obtidos serão sem sentido.

O sistema pode ser recalibrado usando uma fonte com seu fluxo, irradiância ou radiância espectral conhecida. Durante o processo de calibração o arquivo ***.dat** do software será sobrescrito. Antes de calibrar, é recomendado fazer backup do arquivo ***.dat** atual. O processo de calibração usando uma fonte conhecida pode ser realizado clicando no menu **SMS500 Calibration / Calibrate System**. Ao clicar nesse menu uma mensagem será exibida conforme exemplifica a Figura C.3.

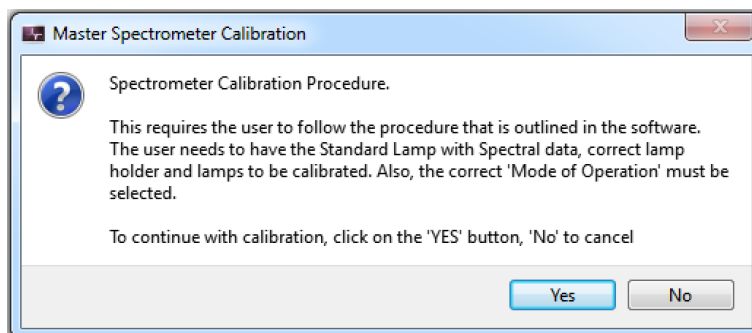


Figura C.3 - Caixa de diálogo para calibração do SMS-500.

Para prosseguir com o processo de calibração clique em **Yes**. Uma caixa de diálogo será exibida para a escolha do arquivo de calibração conforme mostrado na Figura C.4. Selecione o arquivo ***.cal** correspondente a lâmpada de calibração do sistema, e clique em **Abrir**.

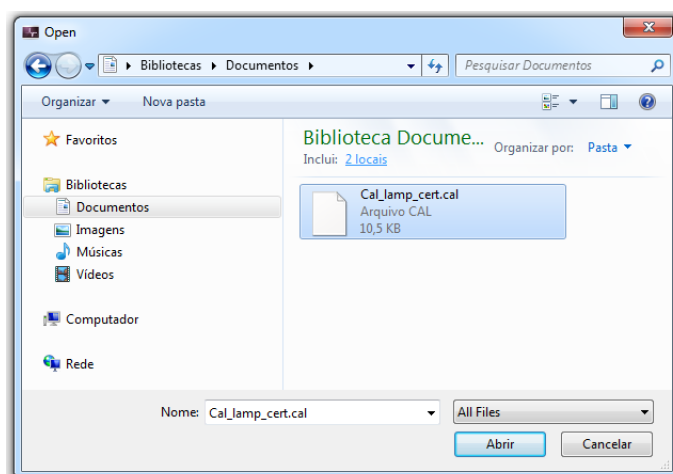


Figura C.4 - Caixa de diálogo para seleção do arquivo de calibração do SMS-500.

Após carregar o arquivo de calibração será exibida a mensagem informando o procedimento que deverá ser realizado para a correta calibração conforme apresentado na Figura C.5. Ligue a lâmpada de calibração e deixe-a ligada por aproximadamente três minutos para estabilização. Após esse intervalo de tempo com a lâmpada ligada, pressione o botão **Yes** para realizar a calibração.

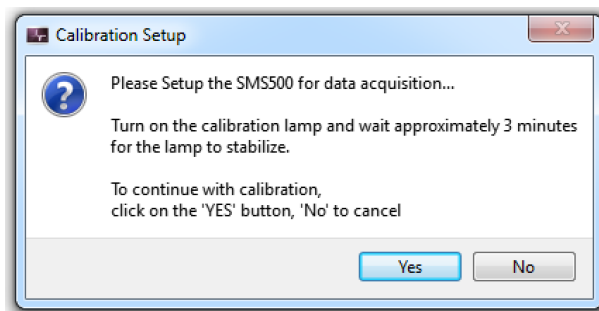


Figura C.5 - Caixa de diálogo com procedimento inicial para calibração do SMS-500.

Ao final da calibração será exibida uma caixa de diálogo igual a apresentada na Figura C.6. Desligue a lâmpada e clique no botão **Yes** para concluir o procedimento. Será exibida a mensagem informando que a calibração foi concluída conforme Figura C.7.

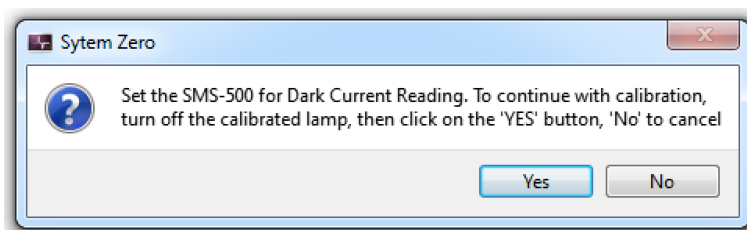


Figura C.6 - Caixa de diálogo com procedimento final para calibração do SMS-500.

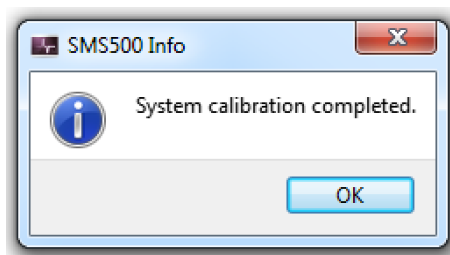


Figura C.7 - Caixa de diálogo informando a finalização da calibração do SMS-500.

Além da calibração usando uma fonte conhecida, também é recomendado realizar o procedimento chamado **System Zero**, que realiza a leitura do ruído de fundo para todos os tempos de integração que o SMS-500 opera. Isso é necessário para a máxima acurácia nas medições.

C.2.2 Opções de configuração do módulo SMS500

Além das duas opções de menu já apresentadas para a calibração do SMS-500, há várias opções de configurações disponíveis diretamente na interface gráfica principal do módulo SMS500.

Start/Stop Scan - inicia ou finaliza a aquisição de dados pelo SMS-500. A quantidade de *scans* pode ser configurada com valor de 1 a 9 999 999 *scans*, ou -1 para quantidade indefinida de *scans*.

Autorange - configura o SMS-500 para encontrar automaticamente o melhor tempo de integração para a aquisição de dados.

Integration Time - configura manualmente o tempo de integração para aquisição de dados.

Samples to Average - configura número de amostras para cálculo da média.

Boxcar Smoothing - seleciona o número de pontos adjacentes usados na função *boxcar smoothing* do SMS-500 para a atenuação do ruído.

Start wavelength - configura o comprimento de onda inicial para aquisição de dados.

Stop wavelength - configura o comprimento de onda final para aquisição de dados.

Use noise reduction - habilita ou desabilita a função *noise reduction*. Com a função habilitada, após cada aquisição de dados, o software adquirirá outro *scan* num tempo de integração maior, e usará as porções do *scan* que não estão saturadas para tentar reduzir o ruído. O tempo de integração maior é determinado pelo tempo de integração base multiplicado pelo valor configurado no fator do *noise reduction*.

Correct for Dynamic Dark - habilita ou desabilita a correção de ruído de fundo.

Save Scan - salva o espectro adquirido juntamente com a configuração do SMS-500 em um arquivo *.txt.

C.3 Módulo LED Driver

O módulo LED Driver pode ser utilizado para controlar a fonte de luz do simulador de estrelas. Por meio desse módulo é possível controlar individualmente a potência de cada canal (grupo de LEDs do mesmo tipo) da fonte de luz espectralmente sintonizável. Cada canal pode ser configurado definindo o valor da potência em níveis digitais, de 0 a 4095.

Nesse módulo também existe uma função específica para coleta da resposta espectral de cada canal em função do nível digital. Essa função é muito importante para a modelagem dos canais e construção da base de dados necessária para a execução do algoritmo Levenberg-Marquardt. A Figura C.8 apresenta a interface gráfica do módulo LED Driver.

Para coletar os dados espectrais dos canais da fonte de luz é possível selecionar o canal de início e de fim, o tipo de atualização do nível digital e o valor de incremento ou decremento do nível digital. A opção **Level Increment/Decrement** configura a forma como os dados são coletados, ou seja, se for selecionado **Level Increment**, um canal por vez será colocado na potência mínima e o espectro adquirido para cada nível digital de 0 a 4095 conforme o incremento definido. No caso de ser selecionado **Level Decrement**, um canal por vez será colocado na potência máxima e o espectro adquirido para cada nível digital de 4095 a 0 conforme decremento definido.

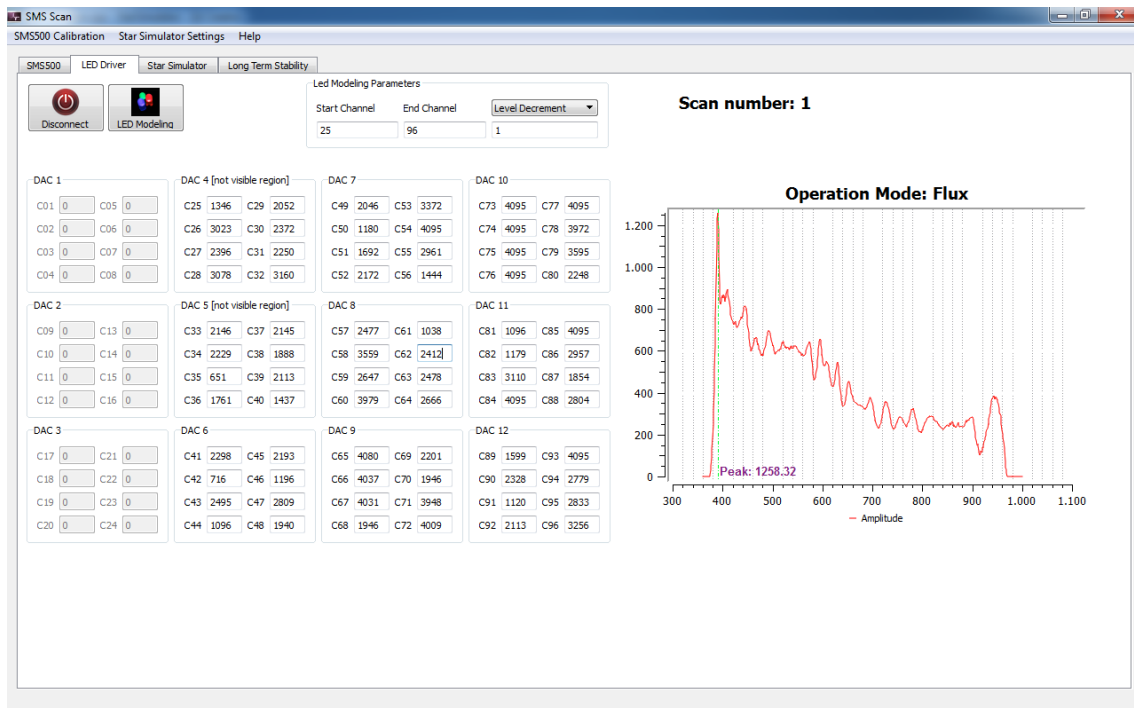


Figura C.8 - Software do Simulador de Estrelas: Módulo LED Driver.

C.4 Módulo Long Term Stability

Este módulo foi implementado para analisar a estabilidade de um determinado espectro a longo prazo. É possível definir o tempo total de execução e o intervalo de tempo que ocorrerá novas coletas de dados.

Antes de usar esse módulo é necessário conectar-se ao SMS-500 e realizar uma primeira coleta de dados. Também é preciso criar uma base de dados para armazenamento das informações coletadas, para isso clique no botão **Create Database**. Após esses procedimentos configure o tempo de execução (**Time to Run**) e intervalo de tempo para novas coletas de dados (**Time Interval**), em seguida clique em **Start Timer** para executar as coletas de dados ao longo do tempo. A Figura C.9 apresenta a interface gráfica do módulo **Long Term Stability**.

Ao final do processo de coleta de dados, é possível exportar os dados para um arquivo ***.txt**, para isso clique no botão **Export All Data**. Também é possível abrir as bases de dados criadas com os software, seja para visualizar as informações ou exportar os dados, para isso clique no botão **Open Database**.

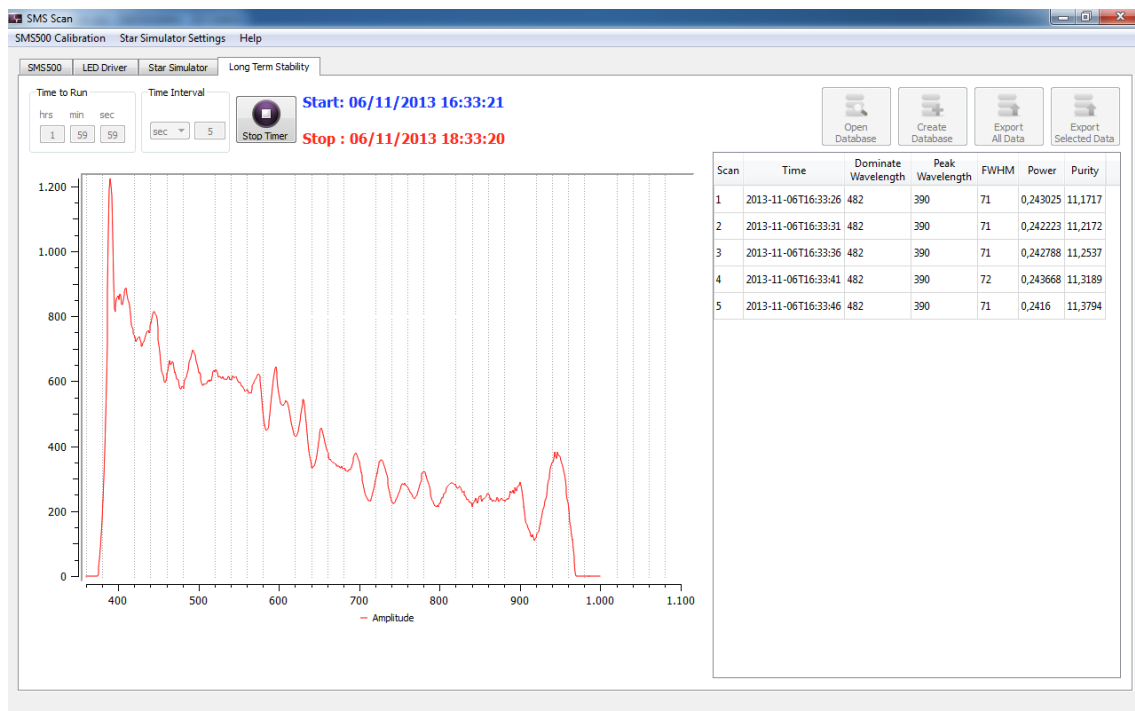


Figura C.9 - Software do Simulador de Estrelas: Módulo Long Term Stability.

C.5 Módulo Star Simulator

Esse módulo utiliza os módulos **SMS500** e **LED Driver** com a finalidade de simular estrelas. A Figura C.10 apresenta a interface gráfica do módulo **Star Simulator**.

As configurações são bem simples:

1. Defina a magnitude e a temperatura da estrela a ser simulada.
2. Selecione o algoritmo para simulação: Levenberg-Marquardt ou Gradiente Descendente.
3. Escolha a solução inicial: pode ser solução aleatória, solução com valores definidos pelo usuário ou usar os valores da última simulação como ponto inicial. A solução inicial também pode ser carregada de um arquivo ***.txt** contendo os níveis digitais para cada canal, para isso clique no menu **Star Simulator Settings / Load initial solution**.

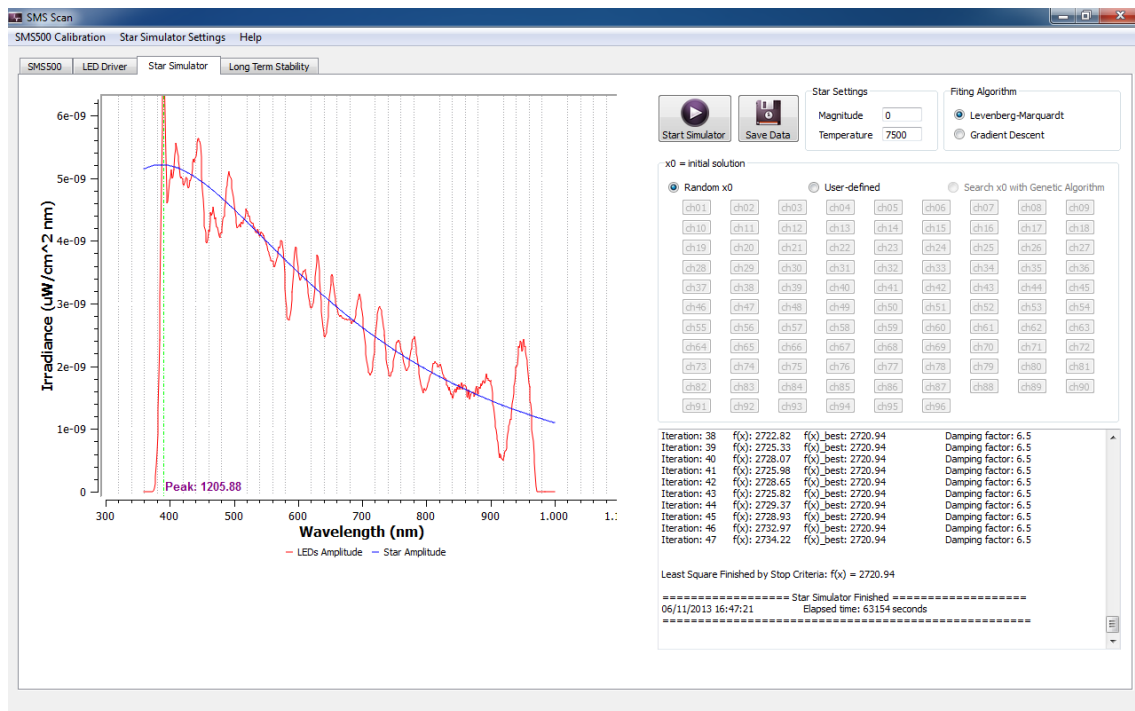


Figura C.10 - Software do Simulador de Estrelas: Módulo Star Simulator.

Após a finalização da simulação da estrela, é possível salvar em um arquivo ***.txt** o espectro conseguido juntamente com os níveis digitais aplicados em cada canal da fonte de luz. Para isso clique no botão **Save Data**.

Para a calibração dos cálculos radiométricos do módulo **Star Simulator** é necessário carregar um arquivo contendo os valores de calibração por comprimento de onda, no intervalo de 360 a 1 000 nanômetros. O arquivo de calibração desse módulo deve conter os valores correspondentes a função de transferência do sistema esfera integradora, *pinhole* e colimador. Para carregar o arquivo clique no menu **Star Simulator Settings / Load transference function table**.

APÊNDICE D - Configuração do ambiente de desenvolvimento

Neste Apêndice, descreve-se de maneira sumária, como foi configurado o ambiente de desenvolvimento do sistema apresentado neste trabalho, fornecendo assim uma base inicial ao leitor que esteja interessado em utilizar o software desenvolvido em trabalhos futuros.

Devido as necessidades do projeto, o ambiente de desenvolvimento foi configurado em uma máquina utilizando Windows 7 64 bits. O framework escolhido para o desenvolvimento do software em linguagem C++ foi a versão *open source* do Qt 5.0 com MinGW 4.7 disponível para download em <http://qt-project.org>.

Para a plotagem dos gráficos apresentados no software desenvolvido foi utilizado a biblioteca Qwt 6.1 disponível para download em <http://qwt.sourceforge.net>.

Para comunicação, controle e aquisição de dados por meio do espectrorradiômetro SMS-500 foi utilizado a DLL **AS5216.dll** fornecida pela SphereOptics, empresa fabricante do espectrorradiômetro.

Para comunicação e controle da fonte de alimentação dos LEDs foi necessário a instalação do driver do chip FT245BL da FTDI disponível para download em <http://www.ftdichip.com/Drivers/D2XX.htm>.

D.1 Instalação e configuração do software SphereOptics SMS-500

Para a correta execução das funções disponíveis pela DLL **AS5216.dll** e para solução de dependências de outras DLLs, se fez necessário instalar o software SMS-500 fornecido juntamente com o espectrorradiômetro. Todos os arquivos necessários estão disponíveis para download em <http://goo.gl/Yk2nML>. Abaixo é apresentado o processo de instalação:

- Baixe e descompacte o arquivo **Labsphere SMS500.iso**.
- Utilize o arquivo **SETUP.exe** para realizar a instalação.
- Baixe e instale o driver para Windows 7 disponível no arquivo **SMS500 driver Win7x86.zip** ou **SMS500 driver Win7x64.zip** conforme a arquitetura do sistema operacional.

Ao executar o aplicativo SMS-500 pela primeira vez será exibido a tela apresentada na Figura D.1. Selecione as opções conforme demonstrado na figura para concluir a instalação corretamente.

Mais informações sobre o espectrorradiômetro estão disponíveis no site do fabricante em <http://www.avantes.com> (Avantes) e <http://www.sphereoptics.de/en> (SphereOptics).

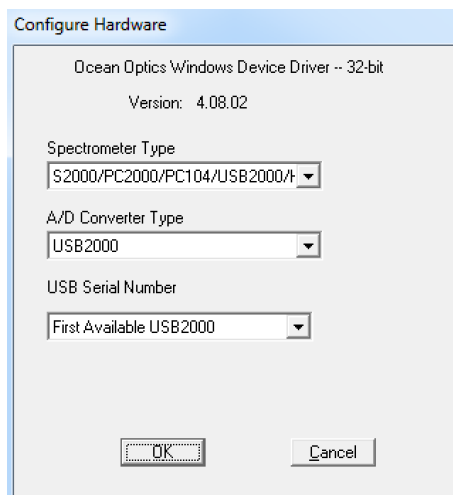


Figura D.1 - Configuração inicial do SMS-500.

D.1.1 Instalação e configuração do Qt 5.0 e Qwt 6.1

A versão open source do framework Qt 5.0 com MinGW 4.7 está disponível para download em <http://qwt.sourceforge.net>. Após baixar e instalar o Qt 5.0, baixe e descompacte a biblioteca Qwt 6.1 no diretório `c:\qwt-6.1.0`. O Qwt está disponível para download em <http://qwt.sourceforge.net>.

Após baixar e descompactar o Qwt, utilize o terminal do Qt acessível pelo menu Iniciar / Todos os Programas / Qt 5.0 / MinGW 4.7 / Qt 5.0 for Desktop, e crie os arquivos de compilação utilizando os seguintes comandos:

```
cd c:\qwt-6.1.0
qmake qwt.pro
```

Compile o código fonte da biblioteca Qwt utilizando o seguinte comando:

```
mingw32-make
```

Instale a biblioteca Qwt com o seguinte comando:

```
mingw32-make install
```

Configure o Qmake para reconhecer o Qwt com o seguinte comando:


```
qmake -set QMAKEFEATURES c:\qwt-6.1.0\features
```

Configure as variáveis de ambientes do Windows, conforme exemplificado nas Figuras D.2 e D.3:

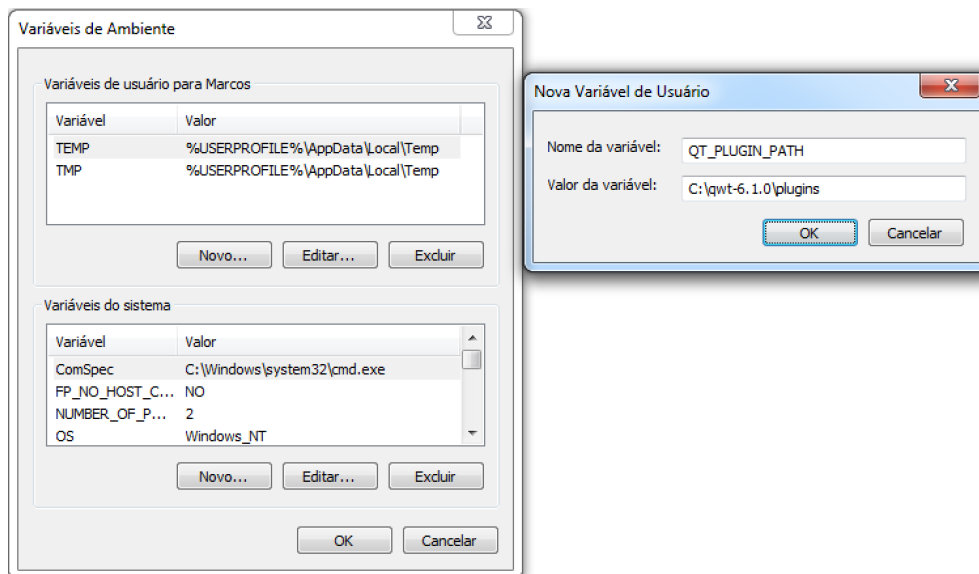


Figura D.2 - Configuração das variáveis de usuário do Windows.

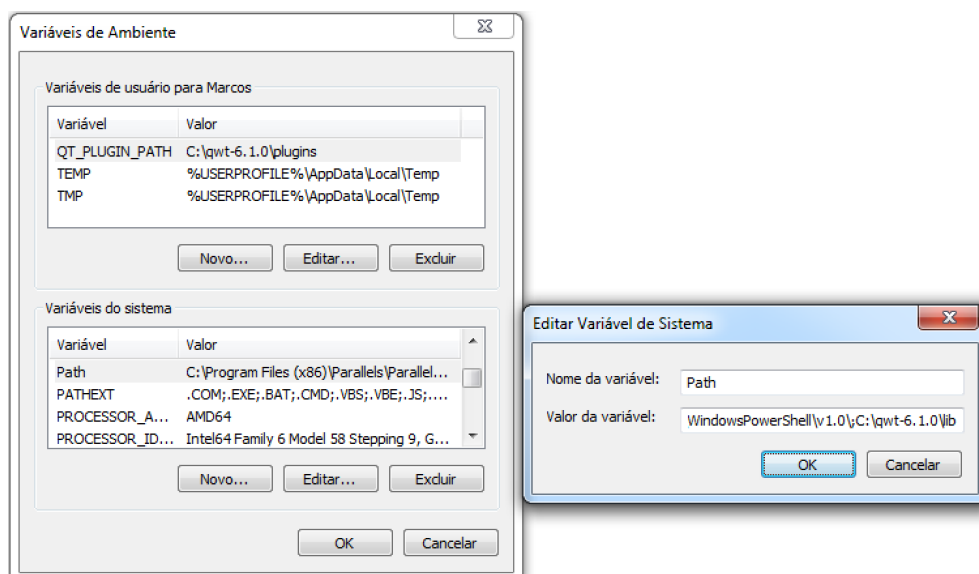


Figura D.3 - Configuração das variáveis de sistema do Windows.

PUBLICAÇÕES TÉCNICO-CIENTÍFICAS EDITADAS PELO INPE

Teses e Dissertações (TDI)

Teses e Dissertações apresentadas nos Cursos de Pós-Graduação do INPE.

Manuais Técnicos (MAN)

São publicações de caráter técnico que incluem normas, procedimentos, instruções e orientações.

Notas Técnico-Científicas (NTC)

Incluem resultados preliminares de pesquisa, descrição de equipamentos, descrição e ou documentação de programas de computador, descrição de sistemas e experimentos, apresentação de testes, dados, atlas, e documentação de projetos de engenharia.

Relatórios de Pesquisa (RPQ)

Reportam resultados ou progressos de pesquisas tanto de natureza técnica quanto científica, cujo nível seja compatível com o de uma publicação em periódico nacional ou internacional.

Propostas e Relatórios de Projetos (PRP)

São propostas de projetos técnico-científicos e relatórios de acompanhamento de projetos, atividades e convênios.

Publicações Didáticas (PUD)

Incluem apostilas, notas de aula e manuais didáticos.

Publicações Seriadas

São os seriados técnico-científicos: boletins, periódicos, anuários e anais de eventos (simpósios e congressos). Contam destas publicações o Internacional Standard Serial Number (ISSN), que é um código único e definitivo para identificação de títulos de seriados.

Programas de Computador (PDC)

São a seqüência de instruções ou códigos, expressos em uma linguagem de programação compilada ou interpretada, a ser executada por um computador para alcançar um determinado objetivo. Aceitam-se tanto programas fonte quanto os executáveis.

Pré-publicações (PRE)

Todos os artigos publicados em periódicos, anais e como capítulos de livros.